

Uncertainty Analysis of Computational Fluid Dynamics Via Polynomial Chaos

Rafael A. Perez

Dissertation submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Aerospace Engineering

Robert W. Walters, Chairman
Jeff Borggaard
Eugene Cliff
Joseph Wang
Richard Barnwell

September 2, 2008
Blacksburg, Virginia

Keywords: Polynomial chaos, stochastic, probabilistic, uncertainty, error
Copyright 2008, Rafael A. Perez

Abstract

The main limitations in performing uncertainty analysis of CFD models using conventional methods are associated with cost and effort. For these reasons, there is a need for the development and implementation of efficient stochastic CFD tools for performing uncertainty analysis. One of the main contributions of this research is the development and implementation of Intrusive and Non-Intrusive methods using polynomial chaos for uncertainty representation and propagation. In addition, a methodology was developed to address and quantify turbulence model uncertainty. In this methodology, a complex perturbation is applied to the incoming turbulence and closure coefficients of a turbulence model to obtain the sensitivity derivatives, which are used in concert with the polynomial chaos method for uncertainty propagation of the turbulence model outputs.

Acknowledgements

I would like express my sincere thanks and appreciation to my advisor, Dr. Robert W. Walters, for his guidance, patience, and for providing me with his work on Polynomial Chaos as a starting point for this thesis, and for providing me with the financial support during my years at Virginia Tech.

I would also like to express my sincere thanks and appreciation to Dr. Jeff Borggaard and Dr. Eugene Cliff for taking the time to guide me in my first year as a graduate student in Virginia Tech. I am also grateful to Dr. Richard Barnwell and Dr. Joseph Wang for their teachings in the field of computational aerodynamics. Special thanks goes to Dr. Eric Nielsen of the NASA Langley Research Center for providing all the free software that I have used during my research work.

Dedication

*to my parents,
to my Grandmother who pass away,
and my wife, Yudith Perez*

Table of Contents

Abstract	i
Acknowledgements	ii
Dedication	iii
Table of Contents	iv
List of Tables	vii
List of Figures	ix
1 Introduction	1
1.1 Error and Uncertainty in CFD Models	1
1.2 Sources of Error and Uncertainty	2
1.3 Objective of the Thesis	3
1.4 Outline of the Thesis	5
2 Background Review	8
2.1 Types and Origins of Uncertainty	8
2.1.1 Inherent Uncertainty	8
2.1.2 Model Form Uncertainty	8
2.1.3 Parameter Uncertainty	9
2.2 Review of Uncertainty Analysis	10
2.2.1 Non-Probabilistic Methods	10
2.2.2 Probabilistic Methods	16
2.2.3 Sampling Based Methods	17
2.2.4 Spectral Methods	22

3	Intrusive Approach: Development and Implementation	25
3.1	Introduction	25
3.2	Development and Implementation	26
3.2.1	Standard PC Euler Equations	26
3.2.2	Compact PC Formulation	28
3.2.3	Compact Notation	31
3.2.4	Implicit PC Formulation	35
3.3	The Application of the PC Euler Equations	38
3.3.1	Implicit PC Boundary Conditions	39
3.3.2	Deterministic Oblique Shock Wave Results	41
3.3.3	Stochastic Oblique Shock Wave Results	42
3.3.4	Deterministic Expansion Wave Results	43
3.3.5	Stochastic Expansion Wave Results	43
3.3.6	Cosine Shaped Airfoil	46
3.3.7	Deterministic Results	48
3.3.8	Stochastic Results	48
4	Non-Intrusive Approach: Development and Implementation	52
4.1	Introduction	52
4.2	Development and Implementation	53
4.2.1	The Galerkin method	53
4.2.2	The Collocation Method	54
4.2.3	The Collocation Method coupled with Sensitivity Analysis	56
4.3	The Application of NIPC methods to a Supersonic Wedge and Expansion Corner	58
4.3.1	Stochastic Oblique Shock Wave Results	59
4.3.2	Stochastic Expansion Wave Results	62
4.4	The Application of NIPC methods to the Onera-M6-Wing Case	65
4.4.1	Grid Convergence Studies for the Onera-M6-Wing Case	66
4.4.2	Deterministic Onera-M6-Wing Results	68
4.4.3	Stochastic Onera-M6-Wing Results	73
5	Turbulence Model Uncertainty Analysis	78
5.1	Introduction and Background	78
5.2	Background Review on Turbulence Modeling	80
5.2.1	S-A Turbulence Model	82
5.2.2	Model Uncertainty Analysis	85
5.3	The Onera-M6-Wing Case	87
5.3.1	The Onera-M6-Wing Sensitivity results	90

5.3.2	Stochastic Onera-M6-Wing Results	91
6	Conclusions and Discussions	94
6.1	Development and Implementation of the Intrusive Method	94
6.2	Development and Implementation of the NIPC Methods	95
6.3	Turbulence Model Uncertainty Analysis	98
6.4	Future Work	100
	Bibliography	102
A		111
A.1	The Definition of the Operator Form, \oplus	111
B		113
B.1	Full Flux Jacobian Matrix in Deterministic Form	113
B.2	Full Flux Jacobian Matrix in Standard PC Form	113
B.3	Full Flux Jacobian Matrix in Operator Form	116
C		117
C.1	Van Leer Split Flux Vector(VLSFV) in Deterministic Form	117
C.2	VLSFV in Standard PC Form	117
C.3	VLSFV in PC Compact Form	120
C.4	VLSFV Jacobian Matrix in Deterministic Form	121
C.5	VLSFV Jacobian Matrix in PC Compact Form	122

List of Tables

1.1	Source of Uncertainty and Error in CFD Simulations - summarized from Oberkampf and Blotter, Ref. [80]	4
2.1	PDF for representing uncertainties in model inputs:	18
2.2	Hermite polynomials of single variable ξ up to 10 orders.	23
4.1	The mean and standard deviation of (P/P_{ref}) obtained with GB-NIPC and MC methods at location $(x = 1.0, z = 0.296394)$ for the wedge problem. The 95% confidence intervals for the standard deviations(StD) are calculated from the MC simulations using the Bootstrap method.	62
4.2	The mean and standard deviation of (P/P_{ref}) obtained with GBNIPC and MC methods at location $(x = 1.0, z = 0.244897)$ for the expansion case. The 95% confidence intervals for the standard deviations(StD) are calculated from the MC simulations using the Bootstrap method.	65
4.3	Grid Level Size	69
5.1	Constants for the S-A model	83
5.2	Scaled sensitivity of the lift coefficient S_{C_L} and the drag coefficient S_{C_D}	88

List of Figures

1.1	Schematic depiction of outline this thesis. The highlighted area indicate the contribution of this thesis	7
3.1	11 × 21 Compression and Expansion Grids	39
3.2	Residual of the deterministic solution on the [11 × 21] Wedge Grid	42
3.3	First-order PC Residuals on the [11 × 21] Wedge Grid	43
3.4	Mean Density and Standard Deviation contours from the first-order PC simulations and 10,000 MC on the [11 × 21] Wedge Grid	44
3.5	Residual of the deterministic solution on the [11 × 21] Expansion Grid	45
3.6	First-Order PC Residuals on the [11 × 21] Expansion Grid	46
3.7	Mean Density and Standard Deviation contours from the first-order PC simulations and 10,000 MC on the [11 × 21] Expansion Grid	47
3.8	The Cosine Shaped Airfoil [65 × 65] Grid	48
3.9	The Deterministic Residual on the Cosine Shaped Airfoil [65 × 65] Grid	49
3.10	The First-Order PC Simulations on the Cosine Shaped Airfoil [65 × 65] Grid	50
3.11	The First-Order PC Residuals on the Cosine Shaped Airfoil [65 × 65] Grid	51
3.12	The First-Order PC Pressure Coefficient on the Cosine Shaped Airfoil [65 × 65] Grid	51
4.1	An unstructured grid with 3265 <i>nodes</i> . The wedge angle is 10°. The Wedge L.E. is 0.05 units	59
4.2	Comparison of the mean and standard deviation of (P/P_{ref}) between GBNIPC and MC Methods. (a)Mean P/P_{ref} obtained via GBNIPC method. (b) Mean P/P_{ref} obtained via MC method. (c) Standard Deviation of (P/P_{ref}) obtained via GBNIPC method. (d) Standard Deviation of (P/P_{ref}) obtained via MC method.	61
4.3	An unstructured grid with 4110 <i>nodes</i> . The deflection angle is 10°. The Expansion L.E. is 0.05 units	63

4.4	Comparison of the mean and standard deviation of (P/P_{ref}) between GBNIPC and MC Methods. (a)Mean P/P_{ref} obtained via GBNIPC method. (b) Mean P/P_{ref} obtained via MC method. (c) Standard Deviation of (P/P_{ref}) obtained via GBNIPC method. (d) Standard Deviation of (P/P_{ref}) obtained via MC method.	64
4.5	A Typical Onera-M6-Wing mesh	67
4.6	Uniform Grid Refinement	70
4.7	The Iterative and Discretization Error	71
4.8	Deterministic Pressure Contours	72
4.9	The PC modes of the Pressure Coefficient(C_p) at Station $y/b = 0.2$, for input random variables, $\overline{AoA} = 3.06$, $CoV = 1\%$ and $\overline{M} = 0.84$, $CoV = 0.5\%$	75
4.10	The PC modes of the Pressure Coefficient(C_p) at Station $y/b = 0.65$, for input random variables, $\overline{AoA} = 3.06$, $CoV = 1\%$ and $\overline{M} = 0.84$, $CoV = 0.5\%$	76
4.11	95% CI of the Mean Pressure Coefficient(C_p) compare with Experimental results at Station $y/b = 0.2$, for input random variables, $\overline{AoA} = 3.06$, $CoV = 1\%$ and $\overline{M} = 0.84$, $CoV = 0.5\%$	77
4.12	95% CI of the Mean Pressure Coefficient(C_p) compare with Experimental results at Station $y/b = 0.65$, for input random variables, $\overline{AoA} = 3.06$, $CoV = 1\%$ and $\overline{M} = 0.84$, $CoV = 0.5\%$	77
5.1	Onera-M6-Wing viscous mesh (409, 135 nodes)	88
5.2	Scaled sensitivities of the lift coefficient S_{C_L}	89
5.3	Scaled sensitivities of the drag coefficient S_{C_D}	89
5.4	Comparison of the lift coefficient C_L between prediction and experiment	93
5.5	Comparison of the drag coefficient C_D between prediction and experiment	93

Chapter 1

Introduction

1.1 Error and Uncertainty in CFD Models

The common practice in CFD analysis and design is to compute a deterministic or single solution on a fixed grid. This practice will be unacceptable in the near future for several reasons. First, it is impossible to infer a level of accuracy or uncertainty from deterministic CFD models that contain errors and uncertainties. Second, all practical CFD models and parameters are known only to a certain level of accuracy or uncertainty. Thirdly, the reliability of CFD analysis and design is reduced when uncertainties and errors are neglected in the CFD models. For these reasons, it is important to develop methods that perform error and uncertainty analysis in CFD models.

A widely known method for performing basic uncertainty analysis for deterministic systems is the standard Monte Carlo(MC) method. The procedure is: (1) sample input random variable from its assumed probability density function(PDF), (2) compute deterministic output for each sampled input value, and (3) determine statistics of the output distribution such as the mean and the variance. However, the main limitations of the MC method is associated with cost and effort. For example, the

MC method typically requires a large number of model runs that would be beyond the reach of current computational power. To date, performing model runs for realistic aerodynamics problems that require the solution of complex flow fields is prohibitively expensive. In fact, this is one of the motivations for the development of alternative methods.

1.2 Sources of Error and Uncertainty

It is important to distinguish between errors and uncertainties associated with the modeling and simulation process. These terms are commonly used interchangeably in the scientific literature, and can be defined in many forms depending on the application. For CFD simulations, the definition given by Oberkampf and Blotter[59] is adopted.

They defined *uncertainty* as,

A potential deficiency in any phase or activity of modeling and simulation process that is due to a lack of knowledge

And *error* is defined as,

A recognizable deficiency in any phase or activity of modeling and simulation that is not due to a lack of knowledge

They also group sources of error and uncertainty in four general categories:

1. Physical modeling
2. Discretization and solution errors
3. Computer round-off error.
4. Programming errors

The physical modeling process is classified as a source of *uncertainty* according to the definition given by Oberkampf and Blotter[59]. Physical modeling uncertainties arise from mathematical model form assumptions, boundary conditions, initial conditions, and data input to a code. In general, data inputs, initial and boundary conditions to a code (*e.g.*, *geometry data*, *free-stream conditions*) are rarely or if at all exactly known.

The discretization and solution process are classified as sources of *error* that can be quantified and reduced using available methods in the literature. Discretization *error* arises from the replacement of the partial differential equations(PDEs) of the physical models, auxiliary models, and boundary conditions by numerical algebraic equations. In Roache[66], discretization *error* is defined as the difference between the exact solution to the discrete equations and analytical solution to the PDEs.

Computer round-off and programming(or user) processes are also sources of *error* that can be minimized or reduced. Computer round-off error arise from finite precision floating-point numbers which can only represent discrete points on the real number line. In Roache[66], computer round-off *error* is defined as the difference between the exact solution to the discrete equations and computer solution. Table 1.1 shows many examples of sources of uncertainty and error in CFD models.

1.3 Objective of the Thesis

The primary objective of this thesis is to develop and implement methods that perform uncertainty analysis of CFD simulations in order to reduce the associated cost and computational effort required with conventional methods such as the MC method.

The second objective of this thesis is to quantify the turbulence model uncertainty in the Spalart-Allmaras model. It is generally believed that turbulence model uncertainty is one of the largest sources of uncertainty in modern Reynolds-Averaged

Table 1.1: Source of Uncertainty and Error in CFD Simulations - summarized from Oberkampf and Blotter, Ref. [80]

<i>Source</i>	<i>Examples</i>
Physical Modeling	Incompressible Flow Inviscid Flow Potential Flow Viscous Flow Chemically Reacting Flow Transitional/Turbulent Flow
Auxiliary Physical Models	Equation of State Thermodynamics properties Transport properties Chemical models, reaction, and rates Turbulence model
Initial & Boundary Conditions	Wall roughness Far-field Free Surface Free-stream Condition Geometry Representation
Discretization & Solution	Truncation error - spatial and temporal Iterative convergence - steady state Iterative convergence - time dependent
Round-Off Error	Finite - precision arithmetic
Programming & User Error	

Navier-Stokes(RANS) simulations, and is a large contributor of the scatter observed between experimental and CFD data [2].

1.4 Outline of the Thesis

Chapter 1 presented a brief introduction to the concepts of error, uncertainty, and its origins in the context of CFD models. It has also discussed the importance of uncertainty analysis of CFD models and the associated limitations with current conventional methods. This chapter summarizes the objectives and outline of this thesis depicted in Figure 1.1.

Chapter 2 presents a thorough review to the types of errors, uncertainties, and its origins in the context of CFD models. It also addresses the main approaches for the representation and propagation of uncertainty associated with the model input and model formulation of CFD models.

Chapter 3 presents the development and implementation of the so called Intrusive method. This chapter addresses the uncertainty in the CFD input parameters that are modeled by replacing all dependent variables and random parameters in the governing equations with their Polynomial Chaos(PC) expansions. It also presents a supersonic wedge flow and expansion corner case study that illustrates the application of the Intrusive method.

Chapter 4 presents the development and implementation of the so called Non-Intrusive methods. This chapter addresses the need to develop Non-Intrusive methods for the purpose of overcoming some of the difficulties associated with the Intrusive method. It also presents the Onera-M6-Wing case study that illustrates an application of Non-Intrusive methods.

Chapter 5 presents the uncertainties that arise in the physical modeling process

for various CFD turbulence models. It describes the methodology for computing turbulence model uncertainty. This chapter uses the Non-Intrusive method to quantify the effect of uncertainties in the closure coefficients of the Spalart-Allmaras turbulence model and wall functions on the aerodynamic coefficients. It also presents the Onera-M6-Wing case study that illustrates the methodology.

Chapter 6 presents the conclusions and discussions of this thesis. This is followed by the bibliography.

Appendix A.1 presents the definition of the operator form. Appendix B.1 presents the full flux Jacobian matrix in deterministic form. Appendix B.2 presents the full flux Jacobian matrix in standard PC form. Appendix B.3 presents the full flux Jacobian matrix in operator form. Appendix C.1 presents the Van Leer split flux vector(VLSFV) in deterministic form. Appendix C.2 presents the VLSFV in standard PC form. Appendix C.3 presents the VLSFV in PC Compact form. Appendix C.4 presents VLSFV Jacobian matrix in deterministic form. Appendix C.5 presents the VLSFV Jacobian matrix in PC compact form.

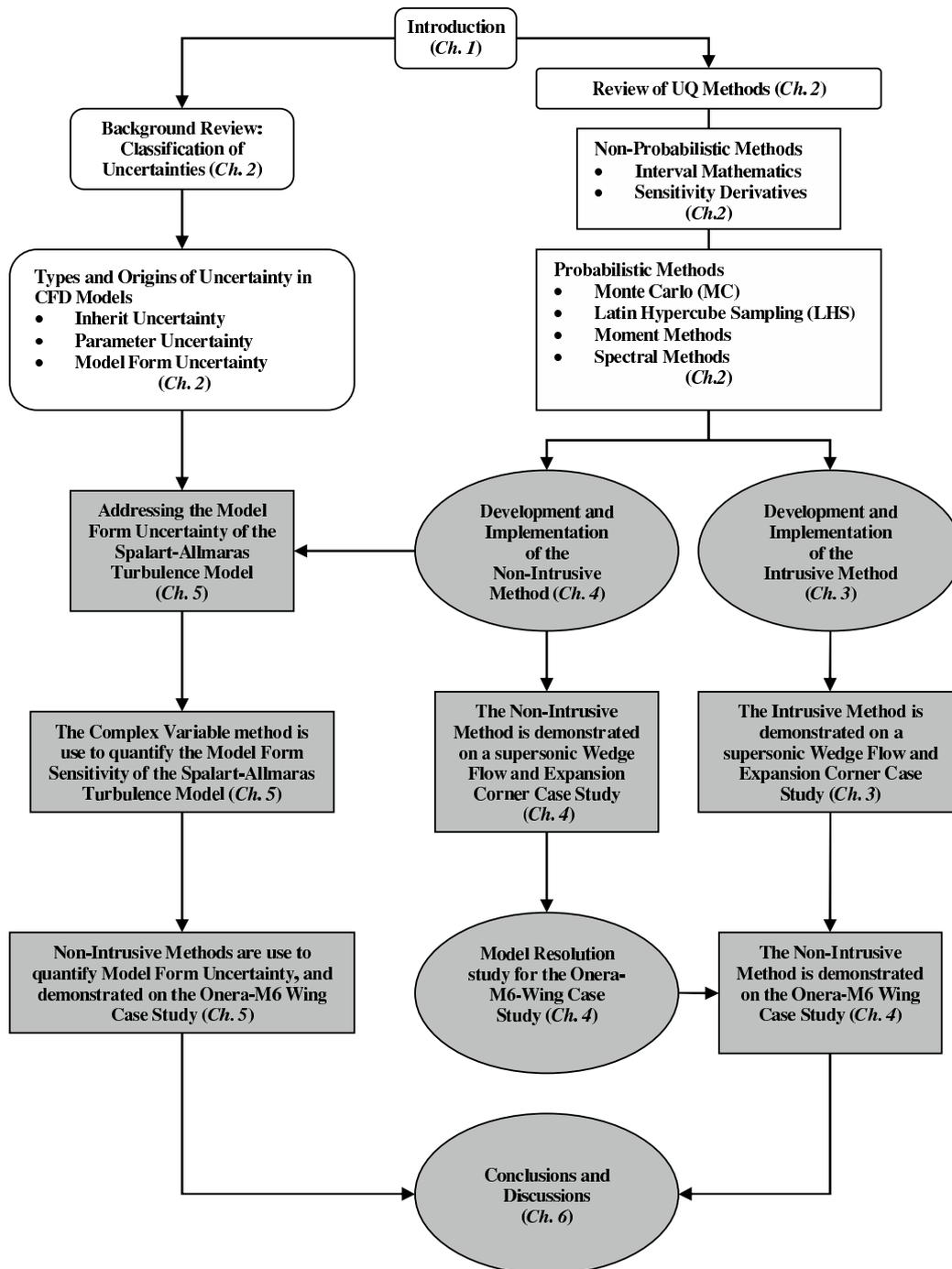


Figure 1.1: Schematic depiction of outline this thesis. The highlighted area indicate the contribution of this thesis

Chapter 2

Background Review

2.1 Types and Origins of Uncertainty

Uncertainty in CFD models can be categorized into aleatoric (or inherent uncertainty) and epistemic (or model form and parameter) uncertainty (see [80]). They are described below.

2.1.1 Inherent Uncertainty

Inherent uncertainty applies to processes or quantities in CFD models that are accepted to be intrinsically variable. Furthermore, processes or quantities that are inherently stochastic arise from the unavoidable unpredictability. For example, chemically reacting gas and transitional turbulent flow are inherently stochastic in nature.

2.1.2 Model Form Uncertainty

In the modeling process, assumptions and simplifications are often required, due to a lack of knowledge. Furthermore, the modeling process leads to mathematical models that are often simplified representations of a phenomena under study. The assumptions and simplifications in these mathematical models often give rise to uncertainty,

and are a key source of uncertainty. It would be possible to treat the mathematical model formulation as an *error* that can be corrected or reduced. However, the exact mathematical model formulation representing a physical phenomena is rarely known, and must be treated according to the definition of *uncertainty*.

The selection of spatial and temporal resolution (*e.g.*, *numerical grid cell size*) in the application of numerical models may also give rise to model uncertainty. For example, Hosder [41] shows the strong interaction between spatial resolution and turbulence model uncertainties. Furthermore, he shows that a finer grid resolution does not necessarily result in a more accurate prediction and reduced turbulence model uncertainty. This is also evident in the *First AIAA CFD Drag Prediction Workshop* [2] where the participants identified the mesh resolution to be inadequate for the CFD predictions. Furthermore, the lack of grid convergence in the CFD predictions led to large variations between the CFD results running similar cases.

The selection of model boundaries may also be a type of simplification that give rise to model uncertainty. For example, any model may have limited boundaries in terms of time, space, number of chemical species, and so on. Other overlooked phenomenon such as rough walls, far-field boundary, free surface, and geometry representation may also play a role in model uncertainty.

2.1.3 Parameter Uncertainty

The true value of model parameters can never be known exactly, and must be treated according to the definition of *uncertainty*. Uncertainty in model parameters originate from significant uncertainties associated with their estimates. For example, uncertainties of parameter measurement involve (1) sampling variability, and (2) systematic bias due to imprecise calibration. Another potential source of parameter uncertainty is the significant large error associated with a small sample size.

2.2 Review of Uncertainty Analysis

The two main approaches for the representation of uncertainties are non-probabilistic (or deterministic) and probabilistic methods.

2.2.1 Non-Probabilistic Methods

In a non-probabilistic approach, uncertainties are often represented by the widest error bounds of model parameters and inputs. The maximum error bounds of model outputs must then be necessarily computed since the probability structure of the model inputs or parameters are not taken into account (see [81]). Furthermore, propagating error using deterministic approaches is based on the assumption that each model input interval contains its entire uncertainty. This assumption may not always be the case. For example, an estimate of the standard deviation of available experimental data may be less (or possibly much less) than the width of the uncertainty model input interval. Two deterministic uncertainty analysis methods (1) Interval Mathematics and (2) propagation of error using sensitivity derivatives are discussed below.

Interval Mathematics

Interval mathematics is used for uncertainty estimation for cases where information about the type of uncertainty in the model parameters is not available. Furthermore, unknown or indeterminate probabilities of model parameters is very often a result of imprecise measurements, and the existence of alternative methods to estimate model parameters. For such cases, interval mathematics uses error bounds to estimate uncertainty. Note that this method does not require information about the type of uncertainty in the parameters (see G. Alefeld and J. Herzberger [4]).

The basic idea in interval mathematics is to compute error bounds on various

model outputs based on the error bounds of the model inputs and parameters. In interval mathematics, every uncertain parameter and uncertain model input are described by an interval that contains an upper and lower limit without a probability structure. Consequently, interval mathematics represent the maximum error bounds (i.e., worst case result). The interval representation of a model parameter, x , defined in terms of the interval midpoint value, \bar{x} , and uncertainty, $\epsilon > 0$, is given by $x = \bar{x}[1 - \epsilon, 1 + \epsilon]$. Note that the lower limit is given by $\bar{x}(1 - \epsilon)$ and the upper limit is given by $\bar{x}(1 + \epsilon)$. Functions of intervals (e.g., intervals of model outputs) are computed using special arithmetic procedures. In order to demonstrate arithmetic operations of interval mathematics, let a and b be given by $[a_l, a_u]$ and $[b_l, b_u]$, where $a_l \leq a_u$ and $b_l \leq b_u$. Then, arithmetic operations of interval mathematics are given by:

$$\begin{aligned}
 a + b &= [a_l + b_l, a_u + b_u], \\
 a - b &= [a_l - b_l, a_u - b_u], \\
 a \cdot b &= [\min(a_l b_l, a_l b_u, a_u b_l, a_u b_u), \max(a_l b_l, a_l b_u, a_u b_l, a_u b_u)], \\
 a/b &= [a_l, a_u] \cdot \left[\frac{1}{b_u}, \frac{1}{b_l} \right]; 0 \notin [b_l, b_u].
 \end{aligned}
 \tag{2.1}$$

An application of interval analysis in the literature includes the treatment of uncertainty in the chemical time scale, τ , for the scalar wave equation with a source term performed by Walters [81]. For this example, he illustrates that different interval results can be obtained for mathematically equivalent pointwise input for different model structural formulation. Further, he shows that this observation is not related to the precision of the computations. He also illustrates that iteration loops required by the numerical algorithms for this case resulted in large error growth without some modification. The fact that (1) probabilistic methods provide more information than interval analysis, (2) many CFD numerical models rely on iteration loops, and (3)

available information about the input probability structure is ignored in interval analysis, detracts from the use of this approach and hence is not recommended in general.

Propagation of Error using Sensitivity Analysis

Propagation of error using sensitivity analysis has been in use for many years in the literature (see e.g. [22], [31], [32], [39], [57], [63], [76]). The objective of sensitivity analysis is to estimate the rate of change in model outputs with respect to changes in model inputs. Furthermore, it can also be used to assess the relative contributions of the model inputs and parameters uncertainty to the model outputs uncertainty.

There are a number of sensitivity analysis methods in the literature (see Isukapalli [47]). The desired method of choice depends upon (a) the type of sensitivity measured, (b) user-defined accuracy, and (c) computational cost. In the CFD community, the desired choice for sensitivity analysis is the local gradient approximation (see Walters [81]). The estimate of the model sensitivity is given by gradients or partial derivatives at a local point in the temporal and spatial domain. If \mathbf{k} is a set of m parameters (k_1, k_2, \dots, k_m) , and \mathbf{u} is a vector of n output variables (u_1, u_2, \dots, u_n) , then the sensitivity, S , is given by,

$$S_{ij} = \frac{\partial u_i}{\partial k_j}. \quad (2.2)$$

If δk_j is the error associated with parameter k_j , then a deterministic approximation to the output error, δu_i , is given by,

$$\delta u_i = \left[\sum_{j=1}^n (S_{ij})^2 \delta k_j \right]^{1/2}, \quad (2.3)$$

where $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, m$.

There are a number of methods for computing sensitivity derivatives. The desired method of choice depends upon the difficulty of implementation and accuracy of the results. The main approaches for computing sensitivity derivatives are (1)

Finite Difference, (2) Complex Variable Formulation, (3) Automatic Differentiation, (4) Discrete Adjoint Method, and (5) the Sensitivity Equation Method (SEM). These methods are discussed below.

Finite Difference

In this approach, the CFD model is treated as a "black box" and sensitivity derivatives are computed by finite differencing perturbed solutions. For example, if f is an output variable of a CFD model with a given perturbation parameter or input value, δx , then the sensitivity of f with respect to a parameter or input variable, x , is given by,

$$\frac{df}{dx} \approx \frac{f(x + \delta x) - f(x - \delta x)}{2 \delta x}. \quad (2.4)$$

Note that the implementation of this central difference scheme is straightforward, and it's theoretically second-order accurate. However, its accuracy depends on the perturbation size. For example, Nielsen [58] shows subtractive cancellation errors arise in finite difference schemes due to finite precision arithmetic regardless of the theoretical order of the scheme. In effect, this limits the step size, δx , that can be used. Note that the limit or optimal step size is not known a priori, and may vary from one design variable to the next. Further, Equation 2.4 requires the function evaluation of two well-converged solutions which in the case of realistic aerodynamic problems may be prohibitively expensive.

Complex Variable Formulation

In this approach, an output CFD function f assumed to be analytic, is expanded in a Taylor series using a complex perturbation parameter or input value, h , as

$$f(x + ih) = f(x) + ih \frac{df}{dx} - \frac{h^2}{2} \frac{d^2 f}{dx^2} - \frac{ih^3}{6} \frac{d^3 f}{dx^3} + \frac{h^4}{24} \frac{d^4 f}{dx^4} + \dots \quad (2.5)$$

Solving this equation for the imaginary part of the function yields,

$$\frac{df}{dx} \approx \frac{Im[f(x + ih)]}{h} + O(h^2). \quad (2.6)$$

The implementation of this method is straightforward. That is, by declaring all variables of a function as complex and applying complex perturbation to the design variable of interest, the sensitivity derivative of a design variable can be obtained by evaluating the imaginary part of a function. Furthermore, Nielsen [58] showed that Equation 2.6 allows true second order accuracy to be achieved, since no subtraction of terms is involved. Moreover, two additional digits of accuracy can be obtained when the step size, h , is reduced by an order of magnitude. However, the drawbacks to this method are associated with the additional cost of performing complex arithmetic, which can be on the order of three times the cost of the original solver.

Automatic Differentiation of Source Code

In this approach, automatic differentiation is applied to a given original source code to generate a corresponding derivative calculating code (see [12], [13], [33], [47] for examples in Fortran). Given the source code, and the definition of the dependent and independent generic variables of interest, automatic differentiation generates a derivative code that is used to estimate the sensitivity and uncertainty of model outputs with respect to model inputs. Note that this method does not require information about the model equation, and the derivative code can be calculated with the accuracy of the machine precision. Although automatic differentiation can be applied to non-smooth functions, the computational cost for real world applications may be prohibitive. Another drawback to this approach, which employs a discretize-then-differentiate scheme, is that mesh derivatives are still not well understood, particularly in cases where meshes are prescribed adaptively (see [77]).

SEM

In this approach, the model equations are differentiated with respect to any generic variable of interest. The subsequent linear sensitivity equations are then discretized

and solved separately from the original model equations. Note that this approach is also termed as the direct approach, and is often applied to cases involving many objectives or constraints with relatively few design variable (see [47], [58]). In [31], Godfrey and Cliff used the SEM for turbulent-flow computations that incorporates an eddy-viscosity model. Further, they showed the solutions to the linear sensitivity equations required approximately 1% of the computational time required to solve the non-linear flow problem. In [14], Borggaard also shows that using the SEM could reduce CPU times by 50 percent or more. However, the drawback to this method is that it requires the modification of the original model equations, and may involve the reformulation of the auxiliary equations. These requirements may be impractical or impossible in terms of the prohibitive amounts of resources that could be required.

Discrete Adjoint Method

In this approach, a user-defined cost function, F , is minimized using the Discrete Adjoint (DA) equations described in Nielsen [57]. The CFD flow equation residuals, \mathbf{R} , and the cost function, F , are first linearized with respect to the flow solution, \mathbf{Q} , and the design variable of interest, \mathbf{D} . After this linearization, a set of linear equations are then solved to find the Lagrange multipliers, λ , given by:

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\right)^T \lambda = - \left(\frac{\partial \mathbf{F}}{\partial \mathbf{Q}}\right)^T, \quad (2.7)$$

where λ is defined as the effect of the flow residual on the cost function:

$$\lambda \equiv \frac{\partial \mathbf{F}}{\partial \mathbf{R}}. \quad (2.8)$$

Once the vector of Lagrange multipliers is known, an iterative scheme for the sensitivity derivatives $\partial \mathbf{Q} / \partial \mathbf{D}$ is derived with Lagrange multipliers operating on the discrete version of the governing flow equations (for details see [58]). This iterative scheme can then be solved efficiently using standard methods. Note the DA method is commonly

used in aerodynamic optimization design problems, and the basic implementation precludes the need to compute the sensitivity derivatives directly (see [81]). However, the direct approach (e.g., SEM or Complex Variable Formulation) may be more appropriate for sensitivity/uncertainty analysis, since the solution of each design variable yields sensitivity information for all of the dependent variables in the flow field (see [58]).

2.2.2 Probabilistic Methods

In the probabilistic approach, uncertainty is represented by the probability of the random event. The probability of a random event occurring can be interpreted in terms of frequency of occurrence of that event, and is determined by the ratio of the number of favorable outcomes to the total number of outcomes. Furthermore, probability is mathematically expressed as a decimal range from a low of 0 (no chance) to a high of 1.0 (certainty). An assessment of these underlying events can be obtained when probability analysis is applied to a collection of data or model parameters and inputs. Furthermore, probabilistic analysis is the prevalent choice for uncertainty analysis of physical systems when estimates of the probability distribution of uncertain model inputs or parameters are available.

There are number of statistical text books that describe the theory of probability. For example, Hafner [35], Papoulis [61], and Tsokos [78] explain the concepts and applications of probabilistic analysis in detail. Uncertainties associated with model inputs and parameters can be quantified by probability distributions, and an estimate of the model output probability distribution can be obtained. Note that this process is comprised of two stages. The first stage involves the determination of the probabilistic distribution of the model inputs and parameters, and model formulation. The second stage involves the propagation of uncertainty through models. These two stages are described below.

Probability distribution of inputs

Probability distributions of model inputs and parameters are estimated via statistical techniques that use available data or a representative number of samples. These techniques can be found in statistical textbooks (see [61]). In cases of limited data, estimates of probability distributions would require expert judgment. For example, a uniform distribution would be chosen for a range of possible values, where all values have an equal likelihood of occurring. Similarly, a normal distribution would be chosen to describe a data set with negligible random and systematic errors. Table 2.1 illustrates some of the probability distributions used for uncertainty analysis.

Propagation of uncertainty through models

The main objective of uncertainty propagation is to compute the probability distribution of model outputs. The output probability distribution can then be used to estimate statistical parameters of interest (e.g., mean and variance of model outputs). The main techniques for propagating uncertainty through models are (1) Sampling Based Methods, and (2) Spectral Methods. These techniques are discussed in the next section.

2.2.3 Sampling Based Methods

Sampling based methods involved running a model at a set of sampled points, and using the model results at the sampling points in order to relate the model inputs and outputs. The advantage of these methods is that the model equation(s) or existing code is treated as a blackbox (*i.e.*, no modification of the model equation(s) or existing code is required). Widely used sampling based methods are: (1) MC, and (2) Latin Hypercube Sampling. These methods are discussed below.

Table 2.1: PDF for representing uncertainties in model inputs:

Distribution	Parameters & Conditions	PDF	Moments
Uniform	a,b	$\frac{1}{b-a}$	Mean = $\frac{a+b}{2}$ Var = $\frac{(b-a)^2}{12}$
Normal	$\mu, \sigma, \sigma > 0$	$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	Mean = μ Var = σ^2 Mode = μ
Lognormal	$\mu, \sigma, \sigma > 0$	$\frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\log(x)-\mu)^2}{2\sigma^2}}$	Mean = $e^{(\mu+\sigma^2/2)}$ Var = $(e^{\sigma^2} - 1)e^{(2\mu+\sigma^2)}$ Mode = $e^{(\mu-\sigma^2)}$
Gamma	$a, b,$ $a > 0, b > 0$	$\frac{1}{\Gamma(a)b^a} x^{a-1} e^{-\frac{x}{b}}$ $x > 0$	Mean = ab Var = ab^2 Mode = $(a-1)b$
Exponential	$\lambda, \lambda > 0$	$\lambda e^{-\lambda x}, x > 0$	Mean = $\frac{1}{\lambda}$ Var = $\frac{1}{\lambda^2}$ Mode = 0
Weibull	a	$ax^{a-1}e^{-x^a} (x \geq 0)$	Mean = $\Gamma(1 + \frac{1}{a})$ Var = $\Gamma(1 + \frac{2}{a}) - \Gamma(1 + \frac{1}{a})^2$ Mode = $(1 - \frac{1}{a})^{\frac{1}{a}}, a \geq 1$
Extreme Value		$e^{-x-e^{-x}}$	Mean = 0 Var = 1 Mode = 0

MC Methods

There are a number of MC methods described in the literature ([19],[38], [50], [70]). They are the most widely used means for uncertainty analysis, and have been applied to fields ranging from chemical engineering [47] to aerospace engineering [82]. They can also be used to solve deterministic problems such as finding the area under a curve [18]. The main advantage of the MC methods is that the model equation(s) or existing code is treated as a blackbox.

The simplest of all MC methods [81], referred to as crude (or basic) MC, involves the sampling of input random variables from their known or assumed PDF, and computing deterministic model output for each of the sample input values. The statistics of the model output distribution can then be determined from the deterministic outputs. The statistics of a distribution can be computed from the definition of the expected value of a function of a random variable, ξ , of $g(\xi)$, namely

$$E[g(\xi)] = \int g(\xi)p(\xi)d\xi, \quad (2.9)$$

where $p(\xi)$ is the PDF of the distribution that describes some event or process. Note the integration domain is supported by the PDF. The mean of the probability distribution is given by

$$\bar{\xi} = E[\xi] = \int \xi p(\xi)d\xi. \quad (2.10)$$

The r^{th} moment about the mean is given by

$$E[(\xi - \bar{\xi})^r] = \int (\xi - \bar{\xi})^r p(\xi)d\xi. \quad (2.11)$$

The coefficients of variance, skewness, and kurtosis are related to the 2^{nd} , 3^{rd} , and 4^{th} moments about the mean, respectively. One of the drawbacks of the MC method is that the convergence of the standard error estimate is relatively slow. For example, the convergence of the MC method [50] to the exact stochastic solution as the number

of samples, $n \rightarrow \infty$, is given by

$$\sigma_{MC} = \frac{\sigma}{\sqrt{n}} . \quad (2.12)$$

Since this method requires a large number of sample or model runs, its not suitable for computationally intensive problems. The time and resources required by this method may be prohibitively expensive for realistic aerodynamics problems.

Latin Hypercube Sampling (LHS)

Substantial computational efficiency over the basic MC method is accomplished by the use of the Modified MC method. The number of necessary solutions is reduced in the Modified MC method compared to the basic MC method by efficient sampling from the input probability distribution. One such widely used Modified MC method is the LHS developed by McKay [54]. In this method, the range for each input uncertain parameter of a model is divided into non-overlapping intervals on the basis of equal probability. Thus, only one value from each interval is selected at random with respect to the PDF in the interval. Note that the LHS method has a smaller variance σ^2 than the MC method [54], hence, the convergence of LHS method is much faster than the MC method. Since the whole parameter space, consisting of all the uncertain parameters, is partitioned into cells of equal probability, random samples are generated from all the ranges of possible values. Consequently, this feature gives insight about the extremes of the probability distributions of the outputs. Note that this feature is not contained in the basic MC method, where there may be cases that does not include the extremes of the random sampling. However, the drawback of the LHS method is that the number of samples or model runs for realistic aerodynamic problems could still be too large and expensive.

Moment Methods

Uncertainty analysis in CFD simulation using moment methods have appeared in the literature (see [43], [44], [45], [76]). Moment methods involves using the truncated Taylor series expanded about the expected value of the inputs. For example, if $u = u(\xi_1, \xi_2)$ is expanded about mean values $(\bar{\xi}_1, \bar{\xi}_2)$, the first-order moment approximation of the Taylor series is,

$$u(\xi_1, \xi_2) = u(\bar{\xi}_1, \bar{\xi}_2) + \frac{\partial u}{\partial \xi_1}(\xi_1 - \bar{\xi}_1) + \frac{\partial u}{\partial \xi_2}(\xi_2 - \bar{\xi}_2).$$

Using Equations 2.9 and 2.11, the expected value and variance are approximately given by,

$$E_{FO}[u(\xi_1, \xi_2)] = u(\bar{\xi}_1, \bar{\xi}_2),$$

$$Var_{FO}[u(\xi_1, \xi_2)] = \left(\frac{\partial u}{\partial \xi_1} \Big|_{\bar{\xi}} \right)^2 \sigma_{\xi_1}^2 + \left(\frac{\partial u}{\partial \xi_2} \Big|_{\bar{\xi}} \right)^2 \sigma_{\xi_2}^2 + 2 \left(\frac{\partial u}{\partial \xi_1} \Big|_{\bar{\xi}} \right) \left(\frac{\partial u}{\partial \xi_2} \Big|_{\bar{\xi}} \right) Covar(\xi_1, \xi_2),$$

where the covariance between the random variables ξ_1 and ξ_2 can be defined in terms of expected values as,

$$Covar(\xi_1, \xi_2) = E[\xi_1 \xi_2 - E(\xi_1)E(\xi_2)].$$

Note that first-order first moment (FOFM) approximation is the deterministic value evaluated at the mean of the inputs, $\bar{\xi}_1$ and $\bar{\xi}_2$. The first-order second moment (FOSM) method requires the computation of sensitivity derivatives (see, for example [63]). For cases involving relatively large variations in the input random variables, increased accuracy of the model output statistics is obtained using higher order moment formulas. This requires the estimation of higher order derivatives which may be impractical in terms of the accuracy and implementation of the method, and the computational resources required. For example, computing higher order derivatives in CFD codes is not well understood and further detracts from the use of this method for uncertainty analysis.

2.2.4 Spectral Methods

Spectral methods have been used to model and propagate uncertainty in stochastic computational simulations by several researchers. Ghanem and Spanos (see [26], [27], [28], [29]) who pioneered spectral representation of uncertainty otherwise known as polynomial chaos expansions, applied this technique to several problems of interest in the structures community. Zang et al. [52] used the PC technique to study uncertainty propagation for a turbulent, compressible nozzle flow. Xiu and Karniadakis [93] analyzed the flow past a circular cylinder and incompressible channel flow by the spectral method, and extended the method beyond the original formulation of Wiener [88] to include a variety of basis functions [94]. In 2003, Walters [82] applied the PC method to a two-dimensional steady-state heat conduction problem for representing geometric uncertainty.

An important concept of spectral representation of uncertainty is that one may decompose a random function (or variable) into separable deterministic and stochastic components. For example, for any generic variable, α^* , with random fluctuations, we can write,

$$\alpha^*(x, y, z, t; \vec{\xi}) = \sum_{i=0}^P \alpha_i(x, y, z, t) H_i(\vec{\xi}), \quad (2.13)$$

where $\alpha_i(x, y, z, t)$ is the deterministic component and $H_i(\vec{\xi})$ is the random basis function corresponding to the i^{th} mode. Effectively, $\alpha_i(x, y, z, t)$ is the amplitude of the i^{th} fluctuation. Here, α^* is assumed to be a function of deterministic independent variables x, y, z, t , and the n -dimensional random variable vector $\vec{\xi} = (\xi_1, \xi_2, \dots, \xi_n)$ which has a specific probability distribution. The discrete sum is taken over the number of output modes, $P \equiv \frac{(n+p)!}{n!p!} - 1$, which is a function of the order of PC, p , and the number of random dimensions, n . For the basis function, multi-dimensional Hermite polynomials are used to span the n -dimensional random space, which was first used by Wiener [88] in his original work known as the homogenous chaos. Many

Table 2.2: Hermite polynomials of single variable ξ up to 10 orders.

k	$H_k(\xi)$
0	1
1	ξ
2	$-1 + \xi^2$
3	$-3\xi + \xi^3$
4	$3 - 6\xi^2 + \xi^4$
5	$15\xi - 10\xi^3 + \xi^5$
6	$-15 + 45\xi^2 - 15\xi^4 + \xi^6$
7	$-105\xi + 105\xi^3 - 21\xi^5 + \xi^7$
8	$105 - 420\xi^2 + 210\xi^4 - 28\xi^6 + \xi^8$
9	$945 - 1260\xi^3 + 378\xi^5 - 36\xi^7 + \xi^9$
10	$-945 + 4725\xi^2 - 3150\xi^4 + 630\xi^6 - 45\xi^8 + \xi^{10}$

other choices are possible for basis functions depending on the type of probability distribution selected for the input uncertainty. For example, Xiu and Karniadakis [93] described other spectral expansions such as Laguerre polynomials with the Exponential distribution, Jacobi polynomials with the Beta distribution, etc. A convenient form of the Hermite polynomials is given by

$$H_k(\vec{\xi}) = e^{\frac{1}{2}\vec{\xi}^T\vec{\xi}}(-1)^k \frac{\partial^k}{\partial \xi_{i_1} \dots \partial \xi_{i_k}} \left(e^{-\frac{1}{2}\vec{\xi}^T\vec{\xi}} \right), \quad (2.14)$$

where $k = 0, 1, \dots, p$ and $i = 0, 1, \dots, n$. Note that p is the order of chaos, n is the number of dimensions, and the row vector $\vec{\xi} = (\xi_{i_1}, \dots, \xi_{i_k})^T$. Table 2.2 gives the Hermite polynomials of one-dimensional random variable (*e.g.*, $n = 1$) up to 10 orders (*i.e.*, $\vec{\xi} = \xi_1 = \xi$). The inner product of two functions $f(\vec{\xi})$ and $g(\vec{\xi})$ is defined by

$$\langle f(\vec{\xi}), g(\vec{\xi}) \rangle = \int_{-\infty}^{\infty} f(\vec{\xi})g(\vec{\xi})p_N(\vec{\xi})d\vec{\xi}, \quad (2.15)$$

where the weight function $p_N(\vec{\xi})$ is an n -dimensional Gaussian distribution with unit variance,

$$p_N(\vec{\xi}) = \frac{1}{\sqrt{(2\pi)^n}} e^{-\frac{1}{2}\vec{\xi}^T\vec{\xi}}. \quad (2.16)$$

The Hermite polynomials form a complete orthogonal set of basis functions in the random space, therefore the inner product of the basis functions is zero with respect to each other,

$$\langle H_i(\vec{\xi}), H_j(\vec{\xi}) \rangle = \langle H_i(\vec{\xi}), H_i(\vec{\xi}) \rangle \delta_{ij}, \quad (2.17)$$

where δ_{ij} is the Kronecker delta function.

The statistics of the distribution for a flow variable at a spatial location can be calculated after the PC coefficients $\alpha_k(x, y, z, t)$ in Equation 2.13 are determined. The mean of the random solution is given by

$$\begin{aligned} E_{PC} [\alpha^*(x, y, z, t; \vec{\xi})] &= \langle \alpha^*(x, y, z, t; \vec{\xi}), H_0(\vec{\xi}) \rangle \\ &= \int_{-\infty}^{\infty} \alpha^*(x, y, z, t; \vec{\xi}) H_0(\vec{\xi}) p_N(\vec{\xi}) d\vec{\xi} \\ &= \alpha_0(x, y, z, t), \end{aligned} \quad (2.18)$$

which indicates that the zeroth mode of the expansions corresponds to the expected value or the mean of $\alpha^*(x, y, z, t; \vec{\xi})$. Similarly, the variance of the distribution can be obtained as,

$$\begin{aligned} Var_{PC} [\alpha^*(x, y, z, t; \vec{\xi})] &= \langle [\alpha^*(x, y, z, t; \vec{\xi}) - \alpha_0(x, y, z, t)]^2, H_k(\vec{\xi}) \rangle \\ &= \int_{-\infty}^{\infty} [\alpha^*(x, y, z, t; \vec{\xi}) - \alpha_0(x, y, z, t)]^2 H_k(\vec{\xi}) p_N(\vec{\xi}) d\vec{\xi} \\ &= \sum_{i=1}^P [\alpha_i^2(x, y, z, t) \langle H_i(\vec{\xi}), H_i(\vec{\xi}) \rangle]. \end{aligned} \quad (2.19)$$

Note the Gaussian estimates of the variance are provided by $k = 1, 2, \dots, n$ modes. All higher modes provide non-Gaussian interactions.

Chapter 3

Intrusive Approach: Development and Implementation

3.1 Introduction

The first section of this chapter presents the development and implementation of the Intrusive method. It describes in detail the derivation of the PC formulation of the Euler equations in the presence of uncertainty. A standard formulation is given for the Full Flux vector and Van Leer split flux vector in local coordinates and their Jacobian matrices. A short-hand compact PC formulation, developed by Walters [85], results in a simplified but equivalent form of the governing equations that are easy to understand and implement.

The second section of this chapter illustrate applications of the Intrusive method. Results have been obtained using stochastic explicit and implicit time integration schemes for the two-dimensional Euler equations of gas dynamics applied to the 2- D unit problems: (1) flow over a wedge at supersonic speed, (2) flow over an expansion corner at supersonic speed, and (3) flow over a cosine airfoil at supersonic speed. Comparisons of first-order PC results show good agreement with the MC simulations

in which 10,000 realizations were obtained. In addition, several convergence and solution contour graphs are shown for the three cases.

3.2 Development and Implementation

In the Intrusive approach, uncertainty in the output variables of CFD models due to uncertainty in model inputs or parameters is represented and propagated by the PC expansions given by Equation 2.13. All dependent variables and random parameters (such as viscosity, thermal conductivity, etc.) or random model inputs (such as free-stream Mach number, geometry, etc.) in the governing equations of the CFD model are replaced with their PC expansions. Projecting each equation onto k^{th} basis, yields $P + 1$ (see Equation 2.13) times the number of deterministic equations. These resultant stochastic linear equations can be solved by the same numerical methods applied to the original deterministic system.

3.2.1 Standard PC Euler Equations

Due to its generality, we work with the integral form of the governing equations

$$\frac{\partial}{\partial t} \int_V \mathbf{Q} dV + \oint_S \mathbf{F} \cdot \hat{n} ds = 0 \quad (3.1)$$

, where the first term is integrated over the volume, V , and the second term is integrated over the closed surface, S .

For the special case of the 2-D Euler equations, $\mathbf{F} = \mathbf{f}\hat{i} + \mathbf{g}\hat{j}$ and

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho e_o \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho u h_o \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho v h_o \end{pmatrix}.$$

Here, the density is ρ , the velocity components are u and v , the pressure is p , and the total internal energy and total enthalpy are given by e_0 and h_0 , respectively. The

components of outward pointing unit normal vector \hat{n} are denoted by n_x and n_y . The cell face surface area is denoted by ds .

The Euler Equations are solved deterministically by discretizing the integral form of the governing equation using a cell-centered finite-volume approach. The integral of the flux on each face of an element is evaluated with the mid-point rule. The spatial accuracy is dictated by interpolating the primitive variable vector $\mathbf{q}(\mathbf{Q})$ from the cell-centers to the cell-faces, where $\mathbf{q} = [\rho, u, v, p]^T$. A standard $\{\phi, \kappa\}$ formulation that allows first-order upwind interpolation and a family of second-order interpolation formulas is used. Discretizing each element in the domain yields,

$$V \frac{\Delta \mathbf{Q}}{\Delta t} + \sum_{i=1}^{N_f} \mathcal{F}_i = 0 \quad (3.2)$$

where,

$$\begin{aligned} V &= \text{element volume} \\ N_f &= \text{number of faces per element} \\ \Delta t &= \text{time step} \\ \mathcal{F} &= \mathbf{F} \cdot \hat{n} \Delta s \end{aligned}$$

In order to obtain the PC equations for the deterministic components, (i.e., the modes of \mathbf{Q}), one simply projects Equation 3.2 onto the r^{th} basis function $\Psi_r(\vec{\xi})$ (for all r). This yields

$$V \frac{\Delta \mathbf{Q}_r \langle \Psi_r(\vec{\xi}), \Psi_r(\vec{\xi}) \rangle}{\Delta t} + \sum_{i=1}^{N_f} \mathcal{F}_{i,r} = 0. \quad (3.3)$$

On the i^{th} face of any element, we denote the projected flux components by

$$[\mathcal{F}_{1,r} \ \mathcal{F}_{2,r} \ \mathcal{F}_{3,r} \ \mathcal{F}_{4,r}]^T$$

where, for example

$$\mathcal{F}_{1,r} \equiv \langle \mathcal{F}_1, \Psi_r(\vec{\xi}) \rangle = \int_{-\infty}^{\infty} \mathcal{F}_1 \Psi_r(\vec{\xi}) p_N(\vec{\xi}) d\vec{\xi} \quad (3.4)$$

and $p_N(\vec{\xi})$ is the input probability distribution function.

3.2.2 Compact PC Formulation

An alternate formulation utilizing operator notation that would be simple to code yet mathematically equivalent to the original formulation was developed and applied to the Euler Equations for explicit time integration by Walters [85]. The definition of the operator will be described by the following simple example. Consider two stochastic scalar variables, a^* and b^* ,

$$a^* = \sum_{i=0}^{N_a} a_i \Psi_i = a_0 \Psi_0 + a_1 \Psi_1 + \cdots + a_{N_a} \Psi_{N_a}, \quad (3.5)$$

$$b^* = \sum_{j=0}^{N_b} b_j \Psi_j = b_0 \Psi_0 + b_1 \Psi_1 + \cdots + b_{N_b} \Psi_{N_b}. \quad (3.6)$$

Note that this can also be written as

$$a^* = a^T \cdot \vec{\Psi}_{N_a} \quad b^* = b^T \cdot \vec{\Psi}_{N_b}, \quad (3.7)$$

where a and b are column vectors containing the deterministic components of a^* and b^* . Likewise, $\vec{\Psi}_{N_a}$ and $\vec{\Psi}_{N_b}$ are column vectors containing the Hermite polynomials (e.g. Ψ_0, Ψ_1, \dots).

Suppose one wants to compute the product of two stochastic variables, and then project the product onto a basis function. Let

$$c^* = a^* b^*. \quad (3.8)$$

Substitute Equations 3.5 and 3.6 into Equation 3.8 to obtain

$$\begin{aligned} c^* &= \left(\sum_{i=0}^{N_a} a_i \Psi_i \right) \left(\sum_{j=0}^{N_b} b_j \Psi_j \right) \\ &= \sum_{i=0}^{N_a} \sum_{j=0}^{N_b} a_i b_j \Psi_i \Psi_j. \end{aligned} \quad (3.9)$$

Given the special case of $N_a = 2$ and $N_b = 1$, c^* has the expansion

$$\begin{aligned}
 c^* = & a_0 b_0 \Psi_0 \Psi_0 + a_0 b_1 \Psi_0 \Psi_1 \\
 & + a_1 b_0 \Psi_1 \Psi_0 + a_1 b_1 \Psi_1 \Psi_1 \\
 & + a_2 b_0 \Psi_2 \Psi_0 + a_2 b_1 \Psi_2 \Psi_1.
 \end{aligned} \tag{3.10}$$

Now consider the simpler term,

$$c = \sum_{i=0}^{N_a} \sum_{j=0}^{N_b} a_i b_j.$$

For demonstration purposes, again let $N_a = 2$ and $N_b = 1$ to expand the above expression to yield,

$$\begin{aligned}
 c = & a_0 b_0 + a_0 b_1 \\
 & + a_1 b_0 + a_1 b_1 \\
 & + a_2 b_0 + a_2 b_1.
 \end{aligned} \tag{3.11}$$

Note that the operation

$$c = a \cdot b^T = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \cdot \begin{bmatrix} b_0 & b_1 \end{bmatrix} = \begin{bmatrix} a_0 b_0 & a_0 b_1 \\ a_1 b_0 & a_1 b_1 \\ a_2 b_0 & a_2 b_1 \end{bmatrix} \tag{3.12}$$

generates all of the terms required to compute the product of two sums. It turns out that a useful PC operation is to simply convert this matrix to a column vector by, in effect, flattening it out. Thus, we define the operator \otimes to be

$$a \otimes b \equiv \text{Flatten} [a \cdot b^T] = \text{Flatten} \begin{bmatrix} a_0 b_0 & a_0 b_1 \\ a_1 b_0 & a_1 b_1 \\ a_2 b_0 & a_2 b_1 \end{bmatrix} = \begin{pmatrix} a_0 b_0 \\ a_0 b_1 \\ a_1 b_0 \\ a_1 b_1 \\ a_2 b_0 \\ a_2 b_1 \end{pmatrix}. \tag{3.13}$$

The original problem can now be written using the definition of this operator as

$$\begin{aligned}
c^* &= \sum_{i=0}^{N_a} \sum_{j=0}^{N_b} a_i b_j \Psi_i \Psi_j \\
&= (a \otimes b)^T \cdot (\vec{\Psi}_{N_a} \otimes \vec{\Psi}_{N_b}).
\end{aligned} \tag{3.14}$$

Equation 3.14 is useful for a PC formulation since the deterministic and random components have been separated, and the operator \otimes is trivial to code. In the *MATHEMATICA*[©] language, it can be defined by

$$a_ \otimes b_ := Flatten[Transpose[\{a\}] \cdot \{b\}], \tag{3.15}$$

where a and b are input lists containing the amplitudes of the modes. Note that when projected onto a basis function, the deterministic component can be taken outside of the integral.

Returning to the example with $N_a = 2$ and $N_b = 1$, the product of two sums expressed in operator form yields,

$$\begin{aligned}
c^* &= (a \otimes b)^T \cdot (\vec{\Psi}_{N_a} \otimes \vec{\Psi}_{N_b}) \\
&= \begin{pmatrix} a_0 b_0 & a_0 b_1 & a_1 b_0 & a_1 b_1 & a_2 b_0 & a_2 b_1 \end{pmatrix} \cdot \begin{pmatrix} \Psi_0 \Psi_0 \\ \Psi_0 \Psi_1 \\ \Psi_1 \Psi_0 \\ \Psi_1 \Psi_1 \\ \Psi_2 \Psi_0 \\ \Psi_2 \Psi_1 \end{pmatrix} \\
&= a_0 b_0 \Psi_0 \Psi_0 + a_0 b_1 \Psi_0 \Psi_1 \\
&\quad + a_1 b_0 \Psi_1 \Psi_0 + a_1 b_1 \Psi_1 \Psi_1 \\
&\quad + a_2 b_0 \Psi_2 \Psi_0 + a_2 b_1 \Psi_2 \Psi_1.
\end{aligned} \tag{3.16}$$

Observe that Equation 3.16 is equivalent to Equation 3.10. Projecting c^* onto the r^{th} basis function yields,

$$\begin{aligned}
\langle c^*, \Psi_r \rangle &= c_r \langle \Psi_r, \Psi_r \rangle \\
&= \sum_{i=0}^{N_a} \sum_{j=0}^{N_b} a_i b_j \langle \Psi_i \Psi_j, \Psi_r \rangle \\
&= (a \otimes b)^T \cdot \int_{-\infty}^{\infty} (\vec{\Psi}_{N_a} \otimes \vec{\Psi}_{N_b}) \Psi_r p_N(\xi) d\xi \tag{3.17}
\end{aligned}$$

$$= (a \otimes b)^T \cdot \langle \vec{\Psi}_{N_a} \otimes \vec{\Psi}_{N_b}, \Psi_r \rangle. \tag{3.18}$$

The last expression (Equation 3.18) is particularly useful when implementing the Intrusive method in operator form. Note that the term $\langle \vec{\Psi}_{N_a} \otimes \vec{\Psi}_{N_b}, \Psi_r \rangle$ is a column vector containing the inner product of user-defined PC combinations as its elements defined by equating Equations 3.18 and 3.17. This vector can be easily computed once the number of user-defined input, internal and output modes are known prior to beginning the iteration process. The notation in Equation 3.18 for the inner product is simplified by introducing

$$\langle \vec{\Psi}_{N_a, N_b, r} \rangle \equiv \langle \vec{\Psi}_{N_a} \otimes \vec{\Psi}_{N_b}, \Psi_r \rangle.$$

In the next section, the deterministic full flux vector, the standard PC formulation of this numerical flux, and the compact formulation are presented using this notation $\langle \vec{\Psi}_{N_a, N_b, r} \rangle$ from this point forward.

3.2.3 Compact Notation

In order to demonstrate the compact PC formulation for the Euler equations, the full flux vector is first written in deterministic form, standard PC form, and compact PC form. The full flux Jacobian matrix is presented in deterministic form, standard PC form, and compact PC form in Appendix B. Appendix C presents the Van Leer split flux vector and Jacobian matrix in all three forms.

PC Full Flux

CFD models use a variety of functions for representing the numerical flux, \mathcal{F} . One of the most common is the full flux vector in local coordinates. Its components are given by:

$$\mathcal{F} = \begin{bmatrix} \rho \bar{u} \\ \rho \bar{u}u + n_x p \\ \rho \bar{u}v + n_y p \\ \rho \bar{u}h_o \end{bmatrix} \Delta s \quad (3.19)$$

where

$$\bar{u} = n_x u + n_y v. \quad (3.20)$$

Substituting the PC expansions (Equation 2.13) into the vector \mathcal{F} (Equation 3.19), and projecting onto the r^{th} basis function yields the PC expansion components for the full flux vector written in the standard form,

$$\begin{aligned} \mathcal{F}_{1,r} &= \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} \rho_k \bar{u}_{ij} \langle \Psi_i \Psi_j \Psi_k, \Psi_r \rangle \\ \mathcal{F}_{2,r} &= \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} \sum_{l=0}^{N_q} \rho_k \bar{u}_{ij} u_l \langle \Psi_i \Psi_j \Psi_k \Psi_l, \Psi_r \rangle + \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} p_j (n_x \Delta s)_i \langle \Psi_i \Psi_j, \Psi_r \rangle \\ \mathcal{F}_{3,r} &= \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} \sum_{l=0}^{N_q} \rho_k \bar{u}_{ij} v_l \langle \Psi_i \Psi_j \Psi_k \Psi_l, \Psi_r \rangle + \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} p_j (n_y \Delta s)_i \langle \Psi_i \Psi_j, \Psi_r \rangle \\ \mathcal{F}_{4,r} &= \frac{\gamma}{\gamma - 1} \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} \bar{u}_{ij} p_k \langle \Psi_i \Psi_j \Psi_k, \Psi_r \rangle \\ &\quad + \frac{1}{2} \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} \sum_{l=0}^{N_q} \sum_{m=0}^{N_q} \rho_k \bar{u}_{ij} q_{lm}^2 \langle \Psi_i \Psi_j \Psi_k \Psi_l \Psi_m, \Psi_r \rangle \end{aligned} \quad (3.21)$$

with

$$q_{lm}^2 = u_l u_m + v_l v_m$$

$$\bar{u}_{ij} = (n_x \Delta s)_i u_j + (n_y \Delta s)_i v_j.$$

Note that Δs has been folded into the definition of \bar{u} for convenience. Utilizing the compact notation described in the previous section, the stochastic flux vector given by Equation 3.21 can be written in the compact form:

$$\begin{aligned} \mathcal{F}_{1,r} &= [\rho \otimes \tilde{u}]^T \cdot \langle \vec{\Psi}_{N_q, N_{g1}, N_q, r} \rangle \\ \mathcal{F}_{2,r} &= [(\rho \otimes \tilde{u}) \otimes u]^T \cdot \langle \vec{\Psi}_{N_q, N_{g1}, N_q, N_q, r} \rangle + (n_x \Delta s \otimes p) \cdot \langle \vec{\Psi}_{N_{g1}, N_q, r} \rangle \\ \mathcal{F}_{3,r} &= [(\rho \otimes \tilde{u}) \otimes v]^T \cdot \langle \vec{\Psi}_{N_q, N_{g1}, N_q, N_q, r} \rangle + (n_y \Delta s \otimes p) \cdot \langle \vec{\Psi}_{N_{g1}, N_q, r} \rangle \\ \mathcal{F}_{4,r} &= \frac{\gamma}{\gamma - 1} [p \otimes \tilde{u}]^T \cdot \langle \vec{\Psi}_{N_q, N_{g1}, N_q, r} \rangle + \frac{1}{2} [(\rho \otimes \tilde{u}) \otimes q_m^2]^T \cdot \langle \vec{\Psi}_{N_q, N_{g1}, N_q, N_q, r} \rangle \end{aligned} \quad (3.22)$$

where

$$\tilde{u} = (n_x \Delta s) \otimes u + (n_y \Delta s) \otimes v,$$

$$q_m^2 = u \otimes u + v \otimes v,$$

N_{int} , user-define internal modes,

N_{g1} , user-define input geometric modes,

$$N_q = \frac{(n+p)!}{n!p!} - 1,$$

$$r = 0, 1, \dots, N_q.$$

Note the similarity between the deterministic full flux vector, Equation 3.19, and the stochastic full flux vector, Equation 3.22. Note the multiplication in Equation 3.19 is replaced by the operation \otimes in Equation 3.22, and scalar inputs become vector arrays. With a little practice, it becomes easy to directly write the compact PC form of the governing equations including the proper definition of the weights (i.e., the inner product vector). Equation 3.22 is written in a much more visually appealing

form by using the operator, \oplus , defined in Appendix A.1. The PC compact full flux vector given in Equation 3.22 can be written in operator form, \oplus , as

$$\begin{bmatrix} \{\rho \otimes \tilde{u}, 0\} \\ \{(\rho \otimes \tilde{u}) \otimes u, n_x \Delta s \otimes p\} \\ \{(\rho \otimes \tilde{u}) \otimes v, n_y \Delta s \otimes p\} \\ \{\frac{\gamma}{\gamma-1}(p \otimes \tilde{u}), \frac{1}{2}(\rho \otimes \tilde{u}) \otimes q_m^2\} \end{bmatrix} \oplus \begin{bmatrix} \{\langle \Psi_{N_q, N_{g1}, N_q, r} \rangle, 0\} \\ \{\langle \Psi_{N_q, N_{g1}, N_q, N_q, r} \rangle, \langle \Psi_{N_{g1}, N_q, r} \rangle\} \\ \{\langle \Psi_{N_q, N_{g1}, N_q, N_q, r} \rangle, \langle \Psi_{N_{g1}, N_q, r} \rangle\} \\ \{\langle \Psi_{N_q, N_{g1}, N_q, r} \rangle, \langle \Psi_{N_q, N_{g1}, N_q, N_q, N_q, r} \rangle\} \end{bmatrix} \quad (3.23)$$

where

$$\tilde{u} = (n_x \Delta s) \otimes u + (n_y \Delta s) \otimes v$$

$$q_m^2 = u \otimes u + v \otimes v.$$

Conversion to Primitive Variables

In the 2-D Euler formulation, the PC expansions of the primitive vector, $\mathbf{q} = \mathbf{q}(\mathbf{Q})$ yields,

$$\mathbf{q}^* \equiv \sum_{i=0}^{N_q} \mathbf{q}_i(x, y) \Psi_i(\xi) = \sum_{i=0}^{N_q} \begin{pmatrix} \rho_i(x, y) \Psi_i(\xi) \\ u_i(x, y) \Psi_i(\xi) \\ v_i(x, y) \Psi_i(\xi) \\ p_i(x, y) \Psi_i(\xi) \end{pmatrix}. \quad (3.24)$$

Note that Equation 3.3 is used to compute the r^{th} component of the conservative vector, $\Delta \mathbf{Q}_r$, and the r^{th} component of the primitive vector, \mathbf{q}_r , is used to update the r^{th} component of the conservative variable, \mathbf{Q}_r . In order to implement this approach, $\Delta \mathbf{Q}_r$ is first converted to $\Delta \mathbf{q}_r$, and then the update step is perform by $\mathbf{q}_r = \mathbf{q}_r + \Delta \mathbf{q}_r$.

This is achieved by the following relation,

$$\begin{aligned}
\Delta\rho_r &= \Delta\rho_r \\
\Delta u_r \langle \Psi_r^2 \rangle &= \rho^{-1} \otimes \Delta\rho u \cdot \langle \Psi_{N_{int}, N_q, r} \rangle - \rho^{-1} \otimes u \otimes \Delta\rho \cdot \langle \Psi_{N_{int}, N_q, N_q, r} \rangle \\
\Delta v_r \langle \Psi_r^2 \rangle &= \rho^{-1} \otimes \Delta\rho v \cdot \langle \Psi_{N_{int}, N_q, r} \rangle - \rho^{-1} \otimes v \otimes \Delta\rho \cdot \langle \Psi_{N_{int}, N_q, N_q, r} \rangle \\
\Delta p_r \langle \Psi_r^2 \rangle &= \frac{\gamma-1}{2} q_m^2 \otimes \Delta\rho \cdot \langle \Psi_{N_q, N_q, N_q, r} \rangle \\
&\quad - (\gamma-1) u \otimes \Delta\rho u \cdot \langle \Psi_{N_q, N_q, r} \rangle \\
&\quad - (\gamma-1) v \otimes \Delta\rho v \cdot \langle \Psi_{N_q, N_q, r} \rangle \\
&\quad + (\gamma-1) \Delta\rho e_0 \langle \Psi_r^2 \rangle.
\end{aligned}$$

3.2.4 Implicit PC Formulation

One of the main mathematical contributions to the present work by the author is the derivation of the stochastic Jacobian matrices necessary for implicit time integration. In order to derive the Euler Implicit formulation, the face fluxes in Equation 3.2 are treated as unknowns at a future time step, $n+1$, which yields,

$$V \frac{\Delta \mathbf{Q}}{\Delta t} + \mathbf{R}^{n+1} = 0, \quad (3.25)$$

where,

$$\mathbf{R}^{n+1} \equiv \left(\sum_{i=1}^{N_f} \mathcal{F}_i \right)^{n+1}.$$

A Newton linearization of \mathbf{R}^{n+1} at \mathbf{R}^n yields,

$$\mathbf{R}^{n+1} = \mathbf{R}^n + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{q}} \right)^n \Delta \mathbf{q}.$$

Substituting this result into Equation 3.25 yields,

$$V \frac{(\Delta \mathbf{Q})^n}{\Delta t} + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{q}} \right)^n \Delta \mathbf{q} = -\mathbf{R}^n. \quad (3.26)$$

Using the fact that the conservative variable \mathbf{Q} is a function of the primitive variable \mathbf{q} or $\mathbf{Q} = \mathbf{Q}(\mathbf{q})$ yields,

$$\begin{aligned}\Delta \mathbf{Q}^n &= \frac{\partial \mathbf{Q}}{\partial \mathbf{q}} \Delta \mathbf{q}^n \\ &= M^n \Delta \mathbf{q}^n,\end{aligned}\tag{3.27}$$

where,

$$M^n = \begin{pmatrix} 1 & 0 & 0 & 0 \\ u & \rho & 0 & 0 \\ v & 0 & \rho & 0 \\ q^2/2 & \rho u & \rho v & \frac{1}{\gamma-1} \end{pmatrix}.$$

Substituting Equation 3.27 into Equation 3.26, a system of linear equations is obtained,

$$\left[\frac{V}{\Delta t} M^n + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{q}} \right)^n \right] \Delta \mathbf{q}^n = -\mathbf{R}^n,\tag{3.28}$$

where $\Delta \mathbf{q}^n$ is the unknown and the update step is performed by $\mathbf{q}^{n+1} = \mathbf{q}^n + \Delta \mathbf{q}^n$.

In order to obtain the PC equations from the deterministic implicit formulation given by Equation 3.28, one simply projects Equation 3.28 onto the r^{th} basis function for all r . This yields a system of stochastic linear equations,

$$\left[\frac{V}{\Delta t} \langle M \Delta \mathbf{q}, \Psi_r^2 \rangle + \frac{\partial \mathbf{R}_r}{\partial \mathbf{q}_r} \Delta \mathbf{q}_r \langle \Psi_r^2 \rangle \right]^n = (-\mathbf{R}_r)^n \langle \Psi_r^2 \rangle.\tag{3.29}$$

Furthermore, in the limit as $\Delta t \rightarrow \infty$ in Equation 3.29, Newton's method is obtained as,

$$\left[\frac{\partial \mathbf{R}_r}{\partial \mathbf{q}_r} \Delta \mathbf{q}_r \langle \Psi_r^2 \rangle \right]^n = (-\mathbf{R}_r)^n \langle \Psi_r^2 \rangle,\tag{3.30}$$

where n denotes the time step, the update is given by $\mathbf{q}_r^{n+1} = \mathbf{q}_r^n + \Delta \mathbf{q}_r^n$ and

$$\mathbf{q}_r = \begin{pmatrix} q_{1,r} \\ q_{2,r} \\ q_{3,r} \\ q_{4,r} \end{pmatrix}, \quad \mathbf{R}_r = \begin{pmatrix} R_{1,r} \\ R_{2,r} \\ R_{3,r} \\ R_{4,r} \end{pmatrix}.$$

Moreover, the term $\left(\frac{\partial \mathbf{R}_r}{\partial \mathbf{q}_r}\right)$ in Equation 3.30 which is an $N_q \times N_q$ (see Equation 3.24) block matrix is given by,

$$\begin{pmatrix} \frac{\partial \mathbf{R}_0}{\partial \mathbf{q}_0} & \frac{\partial \mathbf{R}_0}{\partial \mathbf{q}_1} & \dots & \frac{\partial \mathbf{R}_0}{\partial \mathbf{q}_{N_q}} \\ \frac{\partial \mathbf{R}_1}{\partial \mathbf{q}_0} & \frac{\partial \mathbf{R}_1}{\partial \mathbf{q}_1} & \dots & \frac{\partial \mathbf{R}_1}{\partial \mathbf{q}_{N_q}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{R}_{N_q}}{\partial \mathbf{q}_0} & \frac{\partial \mathbf{R}_{N_q}}{\partial \mathbf{q}_1} & \dots & \frac{\partial \mathbf{R}_{N_q}}{\partial \mathbf{q}_{N_q}} \end{pmatrix}, \quad (3.31)$$

where each element is itself a 4×4 sub matrix.

For the element $\left(\frac{\partial \mathbf{R}_0}{\partial \mathbf{q}_0}\right)$ we have,

$$\begin{pmatrix} \frac{\partial R_{1,0}}{\partial q_{1,0}} & \frac{\partial R_{1,0}}{\partial q_{2,0}} & \frac{\partial R_{1,0}}{\partial q_{3,0}} & \frac{\partial R_{1,0}}{\partial q_{4,0}} \\ \frac{\partial R_{2,0}}{\partial q_{1,0}} & \frac{\partial R_{2,0}}{\partial q_{2,0}} & \frac{\partial R_{2,0}}{\partial q_{3,0}} & \frac{\partial R_{2,0}}{\partial q_{4,0}} \\ \frac{\partial R_{3,0}}{\partial q_{1,0}} & \frac{\partial R_{3,0}}{\partial q_{2,0}} & \frac{\partial R_{3,0}}{\partial q_{3,0}} & \frac{\partial R_{3,0}}{\partial q_{4,0}} \\ \frac{\partial R_{4,0}}{\partial q_{1,0}} & \frac{\partial R_{4,0}}{\partial q_{2,0}} & \frac{\partial R_{4,0}}{\partial q_{3,0}} & \frac{\partial R_{4,0}}{\partial q_{4,0}} \end{pmatrix}. \quad (3.32)$$

The residual contains linear combinations of the numerical flux vectors. Hence, the linearization of the residual contains the Jacobian matrices of the fluxes. For example, for the Van Leer split flux scheme, the Jacobian matrix for the $r = 0$ mode contains the following elements,

$$\begin{pmatrix} \frac{\partial \mathcal{F}_{1,0}^\pm}{\partial \rho_0} & \frac{\partial \mathcal{F}_{1,0}^\pm}{\partial u_0} & \frac{\partial \mathcal{F}_{1,0}^\pm}{\partial v_0} & \frac{\partial \mathcal{F}_{1,0}^\pm}{\partial p_0} \\ \frac{\partial \mathcal{F}_{2,0}^\pm}{\partial \rho_0} & \frac{\partial \mathcal{F}_{2,0}^\pm}{\partial u_0} & \frac{\partial \mathcal{F}_{2,0}^\pm}{\partial v_0} & \frac{\partial \mathcal{F}_{2,0}^\pm}{\partial p_0} \\ \frac{\partial \mathcal{F}_{3,0}^\pm}{\partial \rho_0} & \frac{\partial \mathcal{F}_{3,0}^\pm}{\partial u_0} & \frac{\partial \mathcal{F}_{3,0}^\pm}{\partial v_0} & \frac{\partial \mathcal{F}_{3,0}^\pm}{\partial p_0} \\ \frac{\partial \mathcal{F}_{4,0}^\pm}{\partial \rho_0} & \frac{\partial \mathcal{F}_{4,0}^\pm}{\partial v_0} & \frac{\partial \mathcal{F}_{4,0}^\pm}{\partial v_0} & \frac{\partial \mathcal{F}_{4,0}^\pm}{\partial p_0} \end{pmatrix}. \quad (3.33)$$

Note, that once the element of Equation (3.31) is computed (i.e., $\frac{\partial \mathbf{R}_0}{\partial \mathbf{d}_0}$), the other elements can be computed in a similar fashion.

3.3 The Application of the PC Euler Equations

The problems presented here are an inviscid, steady, two-dimensional supersonic flow of a calorically perfect gas over a wedge and an expansion corner. In these applications, the focus is on the Hermite PC, and its use in propagating uncertainty in the two-dimensional Euler equations. The uncertainty considered herein arises due to uncertainty in a surface definition. The geometric uncertainty was introduced through an angle θ described by a Gaussian PDF. The mean Wedge and Expansion angle is 10° , and the coefficient of variation is 10% (i.e., $\theta(\xi) = \bar{\theta} + \sigma\xi$, where $\bar{\theta} = 10^\circ$, $\sigma = 1$, $\xi = Normal(0, 1)$). In order to represent geometric uncertainty, the PC expansions are substituted into the geometric variables $n_x \Delta s$ and $n_y \Delta s$ which yields,

$$\begin{pmatrix} n_x \Delta s \\ n_y \Delta s \end{pmatrix}^* = \sum_{i=0}^{N_g} \begin{pmatrix} (n_x \Delta s)_i \Psi_i(\xi) \\ (n_y \Delta s)_i \Psi_i(\xi) \end{pmatrix} \quad (3.34)$$

where $i = 0, 1, \dots, N_g$ modes. Typical coarse grids used for these applications are shown in Figure 3.1. The inflow conditions were prescribed by setting the Mach-number, $M_\infty = 3$, and specifying a zero angle-of-attack (AoA). The top and outflow surfaces were extrapolated to first-order, and a tangency boundary condition was prescribed along the bottom surface. These boundary conditions are the same for both the wedge and expansion grids. Note the PC Van Leer split fluxes have been formulated and implemented in the 2-D Euler equations. Since the simulations shown here involve supersonic flow, both the PC Full Flux and the PC Van Leer split fluxes are used.

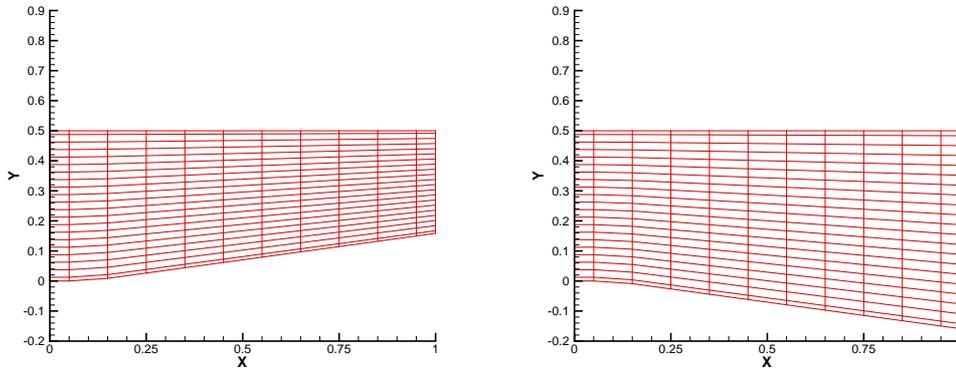


Figure 3.1: 11×21 Compression and Expansion Grids

3.3.1 Implicit PC Boundary Conditions

In order to close the equation set, the appropriate boundary conditions need to be applied. A tangency boundary condition has been implemented on the bottom surface that was defined in terms of values of \mathbf{q} at the first cell-center above the surface. Note the bottom surface is denoted by subscript 1, and the cell-center above by subscript 2. Although many possible choices exist for a tangency boundary condition, the following simple set is imposed,

$$\begin{aligned} \rho_1 &= \rho_2, \\ p_1 &= p_2, \\ \vec{V} \cdot \hat{n} &= \bar{u}_1 = 0, \\ h_{0_1} &= h_{0_2}. \end{aligned}$$

The compact PC version of this boundary condition projected onto the r^{th} basis function is

$$\begin{aligned}
(\rho_r)_1 &= (\rho_r)_2, \\
(p_r)_1 &= (p_r)_2, \\
(u_r)_1 \langle \Psi_r^2 \rangle &= (n_y \otimes \bar{v}_1) \cdot \langle \Psi_{N_{g2}, N_{int}, r} \rangle, \\
(v_r)_1 \langle \Psi_r^2 \rangle &= -(n_x \otimes \bar{v}_1) \cdot \langle \Psi_{N_{g2}, N_{int}, r} \rangle,
\end{aligned}$$

where $\bar{v}_1 = (u \otimes u + v \otimes v)_2^{1/2}$. Note that N_{g2} is the user-define boundary conditions geometric modes, and $r = 0, 1, \dots, Nq$. The PC version of this boundary condition is extended to an implicit PC formulation which yields,

$$\begin{aligned}
\Delta \rho_{r1} \cdot \vec{\delta}_{Nq,l} &= \Delta \rho_{r2} \cdot \vec{\delta}_{Nq,l}, \\
\Delta p_{r1} \cdot \vec{\delta}_{r,l} &= \Delta p_{r2} \cdot \vec{\delta}_{r,l}, \\
\Delta u_{r1} \langle \Psi_r^2 \rangle &= \left[n_{y1} \otimes \left(\frac{\partial \vec{q}}{\partial u} \Delta u + \frac{\partial \vec{q}}{\partial v} \Delta v \right)_2 \right] \cdot \langle \vec{\Psi}_{N_{g2}, N_{int}, r} \rangle, \\
\Delta v_{r1} \langle \Psi_r^2 \rangle &= - \left[n_{x1} \otimes \left(\frac{\partial \vec{q}}{\partial u} \Delta u + \frac{\partial \vec{q}}{\partial v} \Delta v \right)_2 \right] \cdot \langle \vec{\Psi}_{N_{g2}, N_{int}, r} \rangle, \quad (3.35)
\end{aligned}$$

where $q^2 = [u \otimes u + v \otimes v]$ and $l = 0, 1, \dots, Nq$. Computing the terms $(\frac{d\vec{q}}{du})$ and $(\frac{d\vec{q}}{dv})$ in Equation 3.35 involves solving a linear problem of the form

$$[q^*] = \{(q^2)^*\}^{1/2},$$

$$\begin{pmatrix} \Psi_0(\xi_0) & \Psi_1(\xi_0) & \cdots & \Psi_{N_{int}}(\xi_0) \\ \Psi_0(\xi_1) & \Psi_1(\xi_1) & \cdots & \Psi_{N_{int}}(\xi_1) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_0(\xi_{N_{int}}) & \Psi_1(\xi_{N_{int}}) & \cdots & \Psi_{N_{int}}(\xi_{N_{int}}) \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_{N_{int}} \end{pmatrix} = \begin{pmatrix} r_0 \\ r_1 \\ \vdots \\ r_{N_{int}} \end{pmatrix}, \quad (3.36)$$

$$\Psi \vec{q} = \vec{r},$$

where,

$$\begin{pmatrix} r_0 \\ r_1 \\ \vdots \\ r_{N_{int}} \end{pmatrix} = \begin{pmatrix} \left(q_0^2 \Psi_0(\xi_0) + q_1^2 \Psi_1(\xi_0) + \cdots + q_{N_q}^2 \Psi_{N_q}(\xi_0) \right)^{1/2} \\ \left(q_0^2 \Psi_0(\xi_1) + q_1^2 \Psi_1(\xi_1) + \cdots + q_{N_q}^2 \Psi_{N_q}(\xi_1) \right)^{1/2} \\ \vdots \\ \left(q_0^2 \Psi_0(\xi_{N_{int}}) + q_1^2 \Psi_1(\xi_{N_{int}}) + \cdots + q_{N_q}^2 \Psi_{N_q}(\xi_{N_{int}}) \right)^{1/2} \end{pmatrix}.$$

The modes of \vec{q} can be easily found by solving the linear system of equation given by Equation (3.36) for \vec{q} which yields,

$$\vec{q} = \mathbf{\Psi}^{-1} \vec{r}. \quad (3.37)$$

Differentiating Equation (3.37) with respect to u and v , one obtains,

$$\frac{\partial \vec{q}}{\partial u} = \mathbf{\Psi}^{-1} \frac{\partial \vec{r}}{\partial u},$$

$$\frac{\partial \vec{q}}{\partial v} = \mathbf{\Psi}^{-1} \frac{\partial \vec{r}}{\partial v}.$$

3.3.2 Deterministic Oblique Shock Wave Results

Results have been obtained from the implicit time integration scheme applied to the stochastic Euler Equations. The stochastic 2D Euler code space marching option was ran deterministically to steady state by setting the input and output modes to one. The convergence history of the deterministic solution is shown in Figure 3.2. Note that for each i -dim grid points(i.e., grid points normal to the free-stream along the grid), the residual converges quadratically. This was an expected result for a deterministic run that uses Newton's method.

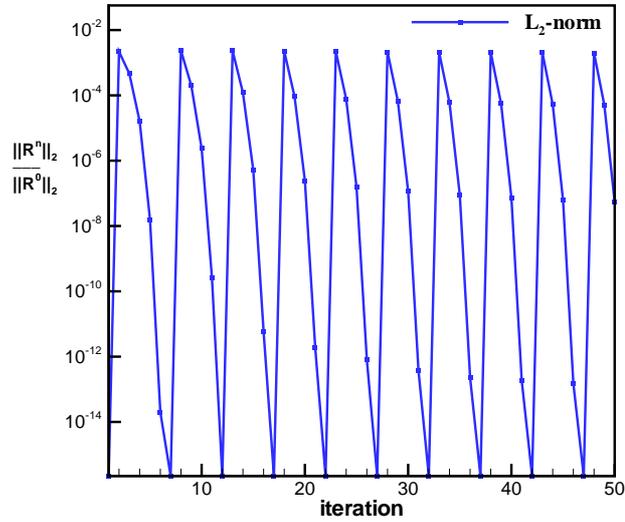


Figure 3.2: Residual of the deterministic solution on the $[11 \times 21]$ Wedge Grid

3.3.3 Stochastic Oblique Shock Wave Results

Geometric uncertainty was introduced through the wedge angle θ . The mean angle was specified to 10° , and the coefficient of variation was 10% (*i.e.*, $\theta(\xi) = \bar{\theta} + \sigma\xi$, where $\bar{\theta} = 10^\circ$, $\sigma = 1$, $\xi = Normal(0, 1)$). For the MC results, 10,000 samples were drawn from a Gaussian PDF. For each of these samples, the deterministic Euler code was executed, and the results were stored. The spectral results shown here are from a first-order PC simulation. The convergence history of the first-order PC is shown in Figure 3.3. Convergence was measured in terms of the L_2 norm of the individual modes and the L_2 norm over all modes. Note that quadratic convergence was obtained for each i -dim grid points for all modes and individual modes as shown in Figure 3.3. Figure 3.4 shows contours of the mean and standard deviation of density from the PC and MC results. Comparisons of first-order PC results show good agreement with the MC simulations. Furthermore, the implicit PC scheme required roughly two orders-of-magnitude less CPU time than the MC method to achieve its result.

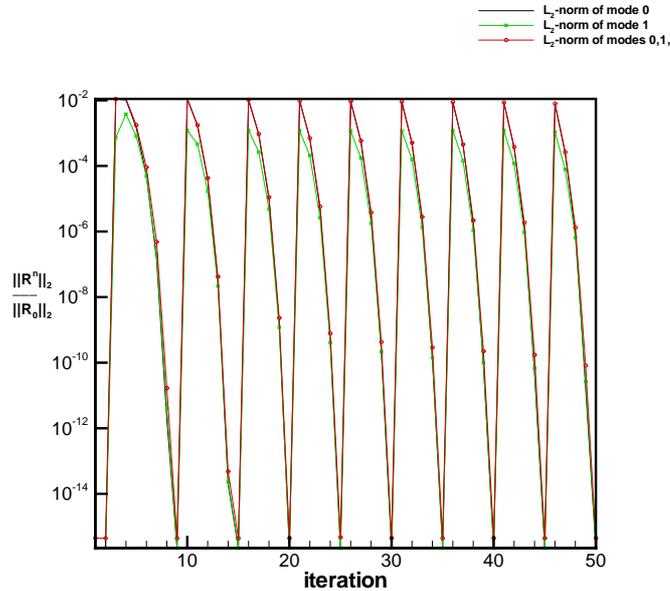


Figure 3.3: First-order PC Residuals on the $[11 \times 21]$ Wedge Grid

3.3.4 Deterministic Expansion Wave Results

The flow over an Expansion corner was also ran deterministically to steady state by setting the input and output modes to one in the stochastic PC Euler code. For this case, the implicit time marching option is used. The convergence history of the deterministic solution is shown in Figures 3.5. Note that quadratic convergence was obtained globally for the time marching scheme. This was an expected result for a deterministic run that uses Newton's method.

3.3.5 Stochastic Expansion Wave Results

Geometric uncertainty was introduced through the expansion angle θ . The mean angle was specified to 10° , and the coefficient of variation was 10% (*i.e.*, $\theta(\xi) = \bar{\theta} + \sigma\xi$, where $\bar{\theta} = 10^\circ$, $\sigma = 1$, $\xi = Normal(0,1)$). For the MC results, 10,000 samples were drawn from a Gaussian PDF. For each of these samples, the deterministic Euler

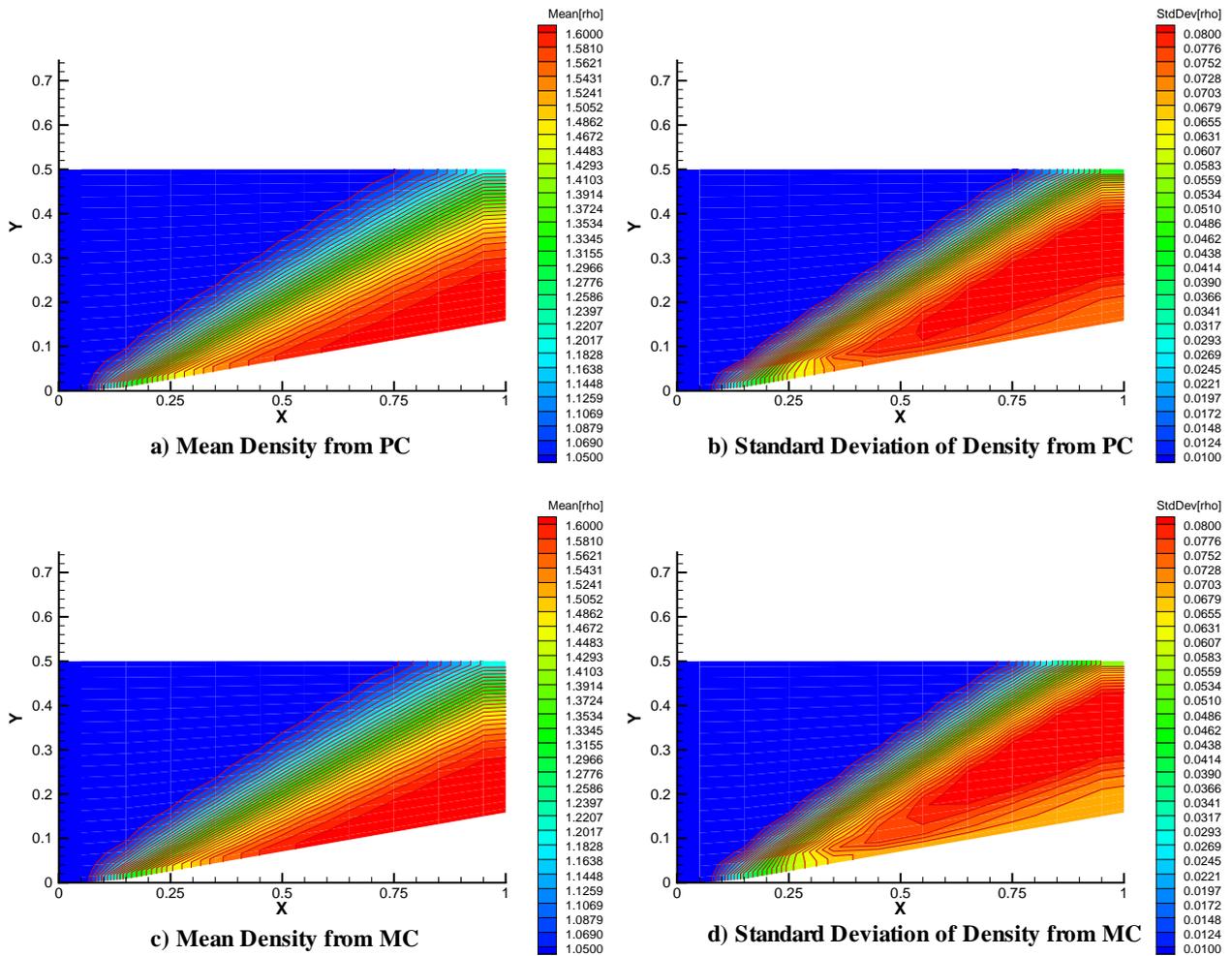


Figure 3.4: Mean Density and Standard Deviation contours from the first-order PC simulations and 10,000 MC on the $[11 \times 21]$ Wedge Grid

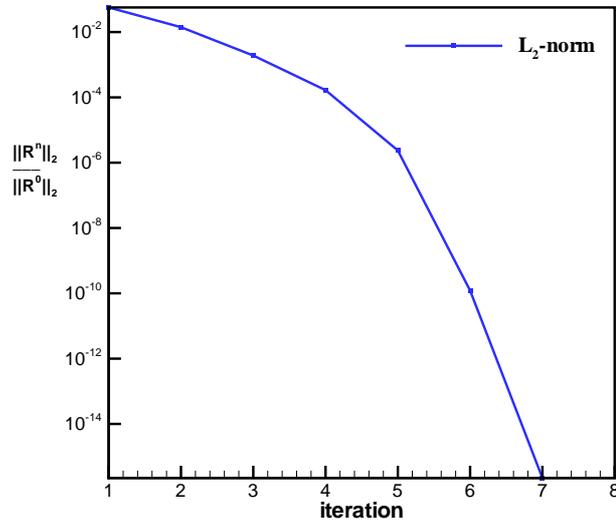


Figure 3.5: Residual of the deterministic solution on the $[11 \times 21]$ Expansion Grid

code was executed, and the results were stored. The spectral results shown here are from the first-order PC simulations. The convergence history of the first-order PC is shown in Figure 3.6. Convergence was measured in terms of the L_2 norm of the individual modes and the L_2 norm over all modes. Note that quadratic convergence was obtained globally. This was an expected result for the stochastic code. Figure 3.7 shows contours of the mean and standard deviation of density from the first-order PC and MC results. Comparisons of the first-order PC results show good agreement with the MC simulations. Furthermore, the implicit PC scheme required roughly two orders-of-magnitude less CPU time than the MC method to achieve its result.

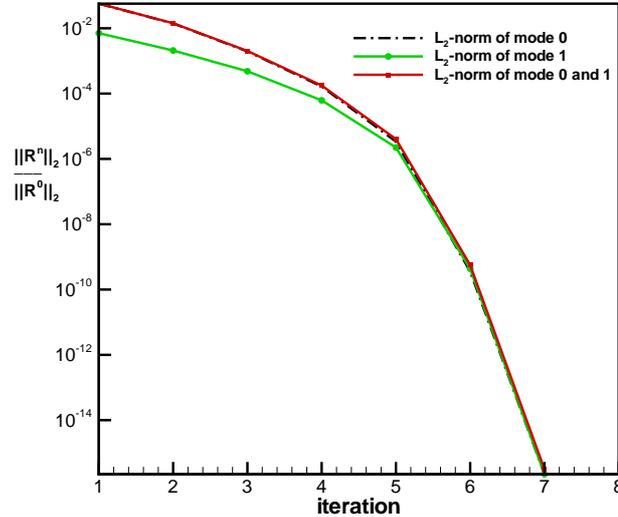


Figure 3.6: First-Order PC Residuals on the $[11 \times 21]$ Expansion Grid

3.3.6 Cosine Shaped Airfoil

The problem presented here is an inviscid, steady, two-dimensional supersonic flow of a calorically perfect gas over a cosine shaped airfoil. In this application, the focus is on the Hermite PC, and its use in propagating uncertainty in the two-dimensional Euler equations. The uncertainty considered herein arises due to uncertainty in surface definition. The geometric uncertainty was introduced through the thickness to chord ratio (t/c) described by a Gaussian PDF. The mean ratio $\overline{t/c}$ was 10% and the coefficient of variation was 1% (*i.e.*, $t/c(\xi) = \overline{t/c} + \sigma \xi$, where $\overline{t/c} = 0.1$, $\sigma = 0.001$, $\xi = Normal(0, 1)$). The PC geometric variables $(n_x \Delta s)^*$ and $(n_y \Delta s)^*$, given by Equation 3.34, is used to represent geometric uncertainty. The grid used in this case is shown in Figure 3.8. The inflow conditions were prescribed by setting the Mach-number, $M_\infty = 3$, at zero AoA. The top and outflow surfaces were extrapolated to first-order. A symmetry boundary condition was prescribed along the surface of the airfoil. Note the PC implicit boundary condition, formulated in section 3.3.1, is also

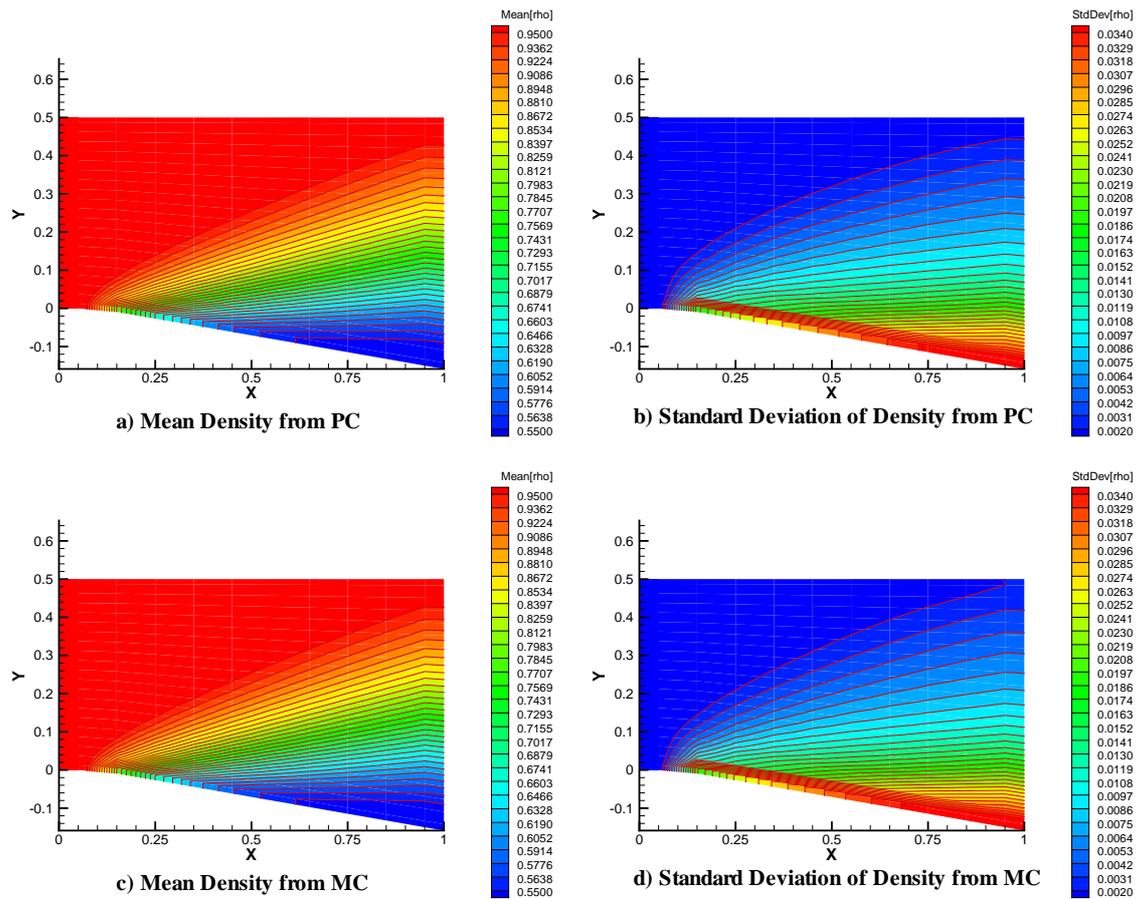


Figure 3.7: Mean Density and Standard Deviation contours from the first-order PC simulations and 10,000 MC on the $[11 \times 21]$ Expansion Grid

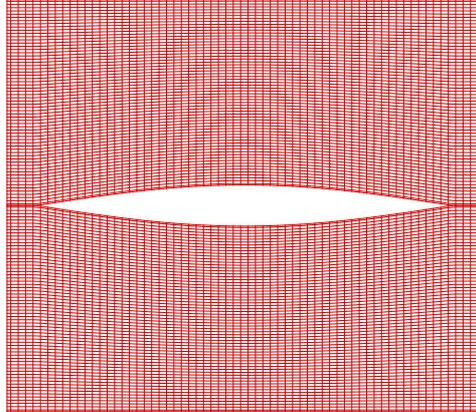


Figure 3.8: The Cosine Shaped Airfoil [65 × 65] Grid

implemented in this application. Since the simulations shown here involves supersonic flow, both the PC Full Flux and the PC Van Leer split fluxes are used.

3.3.7 Deterministic Results

The stochastic 2D Euler code implicit space marching option was ran deterministically by setting the input and output modes to one. The convergence history of the deterministic solution is shown in Figure 3.9. Note that for each i -dim grid points along the grid, the residual converges quadratically.

3.3.8 Stochastic Results

Geometric uncertainty was introduced through t/c . The mean ratio $\overline{t/c}$ was specified to be 10%, and the coefficient of variation was 1%. The spectral results shown here are from the first-order PC simulations. Figure 3.10 shows contours of the mean and standard deviation of pressure from the first-order PC results. The convergence

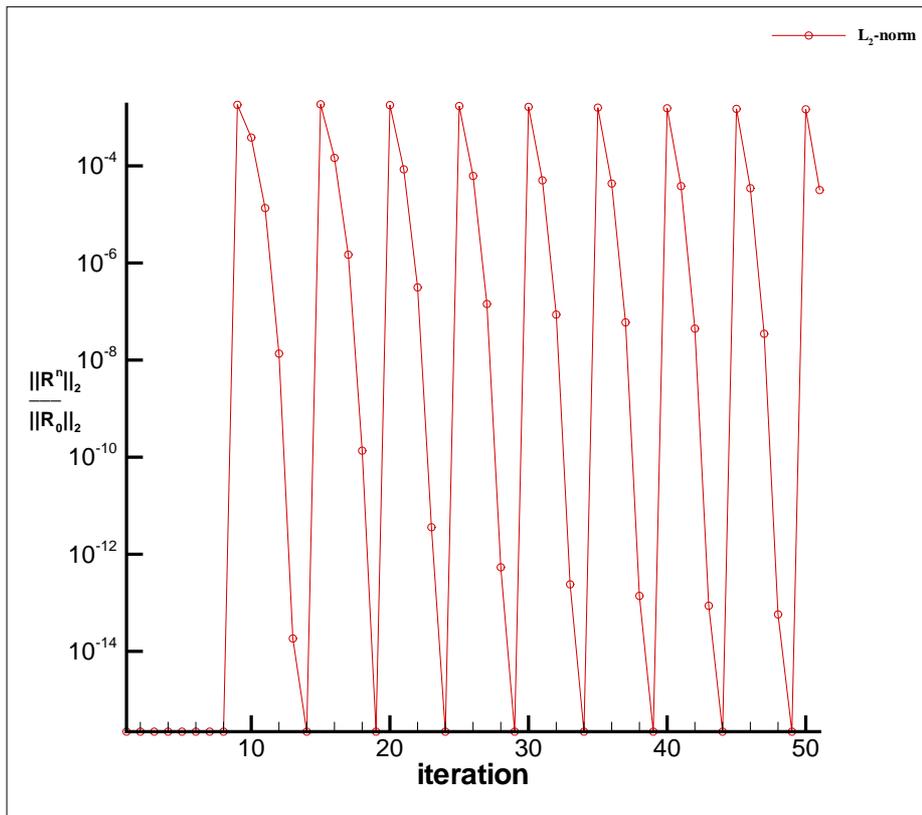


Figure 3.9: The Deterministic Residual on the Cosine Shaped Airfoil $[65 \times 65]$ Grid

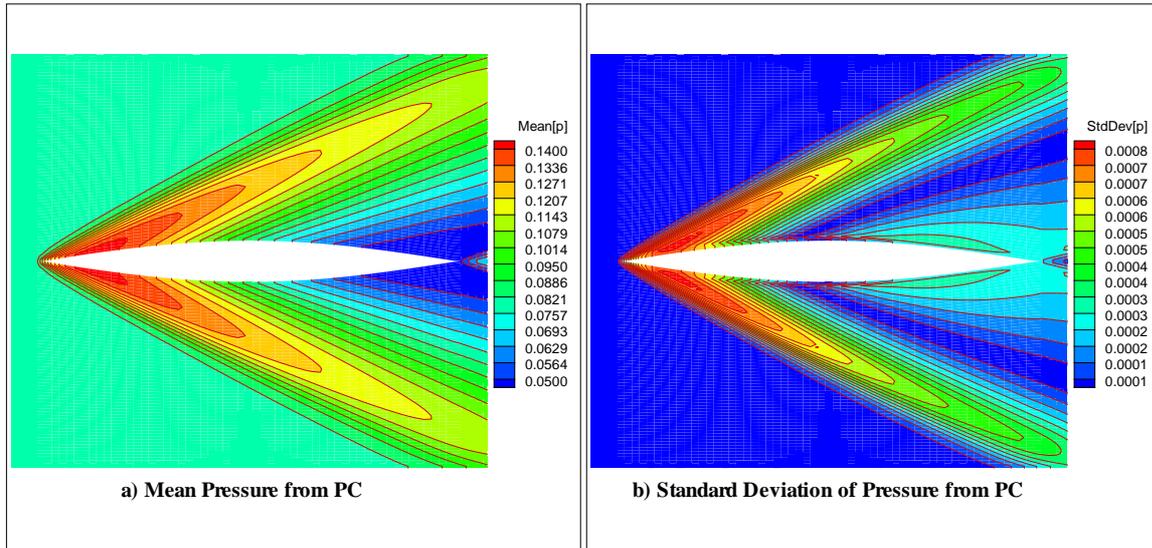


Figure 3.10: The First-Order PC Simulations on the Cosine Shaped Airfoil [65×65] Grid

history of the first-order PC simulations is shown in Figure 3.11. Convergence was measured in terms of the L_2 -norm of the individual modes, and the L_2 -norm over all modes. Note that quadratic convergence was obtained for each i -dim grid points for all modes and individual modes. Figure 3.12 illustrates the pressure coefficient along with the 95% confidence interval(CI) from the first-order PC results and exact solution obtained from compressible flow theory. Note the exact pressure coefficient is not in close agreement with the mean pressure coefficient. This discrepancy is due to lack of grid points near the leading edge.

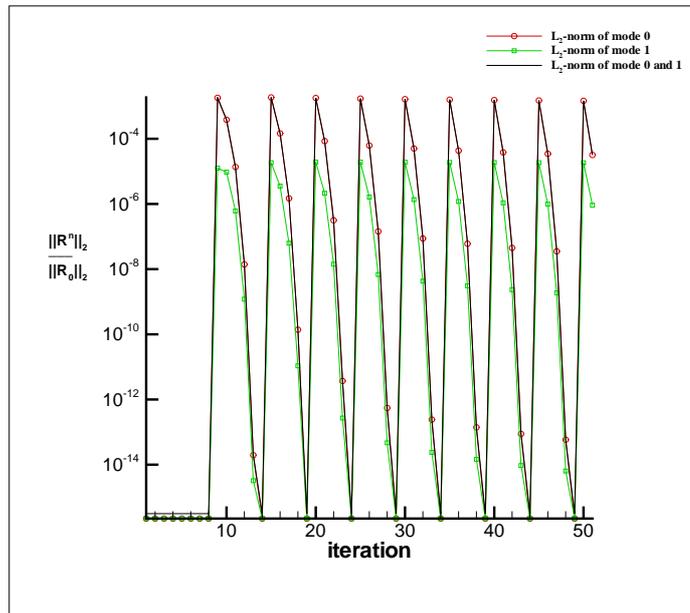


Figure 3.11: The First-Order PC Residuals on the Cosine Shaped Airfoil $[65 \times 65]$ Grid

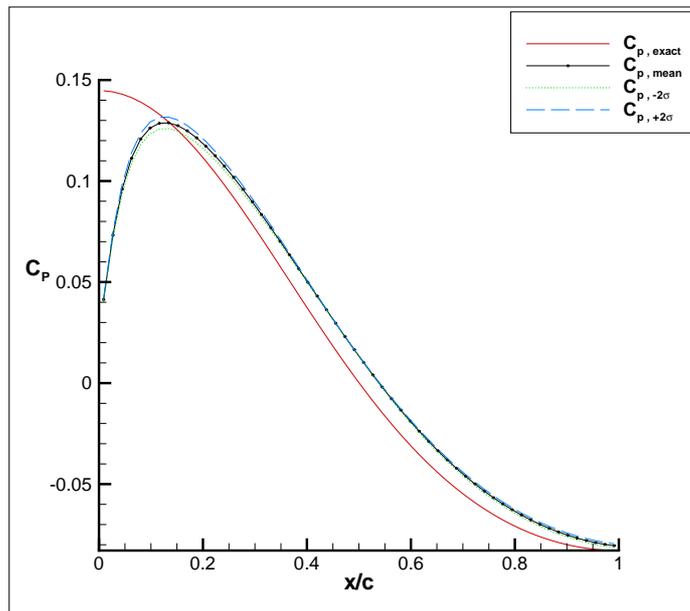


Figure 3.12: The First-Order PC Pressure Coefficient on the Cosine Shaped Airfoil $[65 \times 65]$ Grid

Chapter 4

Non-Intrusive Approach: Development and Implementation

4.1 Introduction

In chapter 3, the Intrusive approach was applied directly to the Euler Equations, and it required modifications in the solver algorithm. In this approach, all dependent variables and random parameters in the Euler equations were replaced with the PC expansions. The resulting equations were then projected onto the k^{th} basis by using the definition of the inner product given by Equation 2.15. These projected equations resulted in $P + 1$ (see Equation 2.13) additional deterministic equations, which were solved by the same conventional numerical technique applied to the original deterministic system. Although straightforward in theory, it was relatively difficult, and time consuming to implement. Hence, it is easily seen that formulating a stochastic CFD code (using the Intrusive approach) capable of handling 3- D , viscous, turbulent flow, around realistic aerospace vehicles would require an enormous modification of an existing CFD deterministic code. However, in the so called Non-Intrusive approach,

no modification to a deterministic code is required. In this new approach, the deterministic code is called as a black box, and uncertainty is modeled and propagated solely by the PC expansions. In fact, the motivation in developing Non-Intrusive PC(NIPC) methods stems from the need to approximate the PC coefficients of the CFD solution without making any modification to the deterministic code.

4.2 Development and Implementation

The NIPC approach was used by Walters [82] to approximate the PC coefficients of the metric terms of a stochastic heat transfer problem with input geometric uncertainty. These input metric PC coefficients were required as an input to the Intrusive PC method. Moreover, Walters [83] used the same approach to determine optimum lift-to-drag ratio for a cosine-shaped airfoil as function of AoA . Similarly, Isukapalli [47] developed a Non-Intrusive method known as the Stochastic Response Surface Method(SRSM) for uncertainty propagation. In the NIPC approach, only the solution, $\alpha^*(x, y, z, t; \vec{\xi})$, is expanded using the appropriate PC basis functions. In this chapter, the focus is on the development and implementation of three NIPC methods based on (1) the Galerkin method, (2) the Collocation method, and (3) the Collocation method coupled with sensitivity analysis. These methods are discussed below.

4.2.1 The Galerkin method

In this method, the solution $\alpha^*(x, y, z, t; \vec{\xi})$ is projected onto the k^{th} basis by using the definition of the inner product(see Equation 2.15) and the property of orthogonality given by,

$$\alpha_k(x, y, z, t) = \frac{\langle \alpha^*(x, y, z, t; \vec{\xi}), \Psi_k(\vec{\xi}) \rangle}{\langle \Psi_k^2(\vec{\xi}) \rangle}. \quad (4.1)$$

In Equation 4.1, the denominator of the PC coefficients is independent of the response α^* , and can therefore be pre-computed. The integral of the inner product in the numerator can be estimated using Gauss quadrature by,

$$\alpha_k(x, y, z, t) \approx \sum_{m_1=1}^m \cdots \sum_{m_n=1}^m \alpha^*(x, y, z, t; x_{m_1}, \cdots, x_{m_n}) \frac{\Psi_k(x_{m_1}, \cdots, x_{m_n})}{\langle \Psi_k^2 \rangle} \prod_{i=1}^n w_{m_i}, \quad (4.2)$$

where (x_{m_i}, w_{m_i}) are the integration points and weights along each stochastic direction, and m is the total number of integration points used in a single dimension. The quadrature in Equation 4.2 is exact when the integrand is a polynomial of degree of $(2m - 1)$ or less. Thus, the PC coefficients can be exactly estimated if the solution is expanded by a PC of degree less than or equal to $(2m - 1)/2$. Note the solution $\alpha^*(x, y, z, t; \vec{\xi})$ assumes the functional form of the PC expansions. However, the degree of departure of the PC expansions depends on the accuracy of the PC coefficients. Although Gauss quadrature is a very efficient method for numerical integration, for practical purposes, it is limited to cases with a few number of random variables. For example, a problem with n number of random variables, the total number of deterministic solutions N_d required for an exact estimation is $N_d \geq (2p - 1)^n$, where p is the order of the chaos.

4.2.2 The Collocation Method

The first step in the formulation of this method is to write down the PC expansions of the solution given by,

$$\alpha^*(x, y, z, t; \vec{\xi}) = \sum_{i=0}^P \alpha_i(x, y, z, t) \Psi(\vec{\xi})$$

where $P = \frac{(n+p)!}{n!p!} - 1$, n is the number of random variables, and p the order of the chaos. The second step is to select $P+1$ collocation points of the random variables and evaluate the code for each of these sample values. The deterministic modal response

$\alpha_i(x, y, z, t)$ is then solved using the computed sample solutions $\alpha^*(x, y, z, t; \vec{\xi}_i)$ for each of the pre-selected collocation points $\vec{\xi}_i$ by,

$$\begin{bmatrix} \Psi_0(\vec{\xi}_0) & \Psi_1(\vec{\xi}_0) & \cdots & \Psi_P(\vec{\xi}_0) \\ \Psi_0(\vec{\xi}_1) & \Psi_1(\vec{\xi}_1) & \cdots & \Psi_P(\vec{\xi}_1) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_0(\vec{\xi}_P) & \Psi_1(\vec{\xi}_P) & \cdots & \Psi_P(\vec{\xi}_P) \end{bmatrix} \begin{pmatrix} \alpha_0(x, y, z, t) \\ \alpha_1(x, y, z, t) \\ \vdots \\ \alpha_P(x, y, z, t) \end{pmatrix} = \begin{pmatrix} \alpha^*(x, y, z, t; \vec{\xi}_0) \\ \alpha^*(x, y, z, t; \vec{\xi}_1) \\ \vdots \\ \alpha^*(x, y, z, t; \vec{\xi}_P) \end{pmatrix} \quad (4.3)$$

$$\alpha_i = [\Psi_P]^{-1} \alpha^*(\vec{\xi}_i). \quad (4.4)$$

The collocation points are selected based on the Orthogonal Collocation Method(OCM) suggested by Villadsen and Michelsen [79]. The orthogonal collocation points correspond to the roots of the polynomial of one degree higher than the order of the PC expansion. For the special case of one-dimensional random variable problem, OCM gives the same result as the Galerkin method [79], and is regarded as the optimal selection. For multi-dimensional random variable problems, the collocation points are selected such that the overall distribution is symmetric with respect to the origin, and takes on values of zero or one of the roots of the higher order polynomial. Although this method is not unique for the solution of PC coefficients of multi-random variables, the behavior of the PC coefficients is captured reasonably well in regions of high probability. For example, the exclusion of the origin as a collocation point could potentially lead to a poor estimation when the origin corresponds to the region of highest probability. Furthermore, singularities can be avoided in Equation 4.3, as the collocation points are selected at the roots of the higher order polynomials.

4.2.3 The Collocation Method coupled with Sensitivity Analysis

The first step in the formulation of this method is to write down the PC expansions of the solution given by,

$$\alpha^*(x, y, z, t; \vec{\xi}) = \sum_{i=0}^P \alpha_i(x, y, z, t) \Psi(\vec{\xi}). \quad (4.5)$$

The second step is to differentiate Equation 4.5 with respect to $Z_i(\xi_i)$ using the chain rule which yields,

$$\left\{ \frac{\partial \alpha^*(x, y, z, t; \vec{\xi})}{\partial Z_i(\xi_i)} \right\} \left\{ \frac{\partial Z_i(\xi_i)}{\partial \xi_i} \right\} = \sum_{i=0}^P \alpha_i(x, y, z, t) \frac{\partial \Psi_i(\vec{\xi})}{\partial \xi_i}, \quad (4.6)$$

where $Z_i(\xi_i)$ are input random variables to a CFD model. Consider two input random variables to a CFD model given by,

$$Z_1(\xi_1) = \mu_1 + \sigma_1 \xi_1$$

$$Z_2(\xi_2) = \mu_2 + \sigma_2 \xi_2$$

where ξ_1 and ξ_2 are two independent identically distributed normal random variables.

The terms $\frac{\partial Z_i(\xi_i)}{\partial \xi_i}$ are then easily computed as,

$$\frac{\partial Z_1(\xi_1)}{\partial \xi_1} = \sigma_1$$

$$\frac{\partial Z_2(\xi_2)}{\partial \xi_2} = \sigma_2.$$

The third step is to select collocation points of the random variables and evaluate the code for each of these sample values. The deterministic modal response $\alpha_i(x, y, z, t)$ is then solved using the computed sample solutions $\alpha^*(x, y, z, t; \vec{\xi}_j)$ and

$\frac{\partial \alpha^*(x, y, z, t; \vec{\xi})}{\partial Z_i(\xi_i)}$ for each of the pre-selected collocation points by,

$$\begin{bmatrix}
 \Psi_0(\vec{\xi}_0) & \Psi_1(\vec{\xi}_0) & \cdots & \Psi_P(\vec{\xi}_0) \\
 \frac{\partial \Psi_0(\vec{\xi}_0)}{\partial \xi_1} & \frac{\partial \Psi_1(\vec{\xi}_0)}{\partial \xi_1} & \cdots & \frac{\partial \Psi_P(\vec{\xi}_0)}{\partial \xi_1} \\
 \vdots & \vdots & \ddots & \vdots \\
 \frac{\partial \Psi_0(\vec{\xi}_0)}{\partial \xi_n} & \frac{\partial \Psi_1(\vec{\xi}_0)}{\partial \xi_n} & \cdots & \frac{\partial \Psi_P(\vec{\xi}_0)}{\partial \xi_n} \\
 \Psi_0(\vec{\xi}_1) & \Psi_1(\vec{\xi}_1) & \cdots & \Psi_P(\vec{\xi}_1) \\
 \frac{\partial \Psi_0(\vec{\xi}_1)}{\partial \xi_1} & \frac{\partial \Psi_1(\vec{\xi}_1)}{\partial \xi_1} & \cdots & \frac{\partial \Psi_P(\vec{\xi}_1)}{\partial \xi_1} \\
 \vdots & \vdots & \ddots & \vdots \\
 \frac{\partial \Psi_0(\vec{\xi}_1)}{\partial \xi_n} & \frac{\partial \Psi_1(\vec{\xi}_1)}{\partial \xi_n} & \cdots & \frac{\partial \Psi_P(\vec{\xi}_1)}{\partial \xi_n} \\
 \vdots & \vdots & \ddots & \vdots \\
 \vdots & \vdots & \ddots & \vdots \\
 \Psi_0(\vec{\xi}_N) & \Psi_1(\vec{\xi}_N) & \cdots & \Psi_P(\vec{\xi}_N) \\
 \frac{\partial \Psi_0(\vec{\xi}_N)}{\partial \xi_1} & \frac{\partial \Psi_1(\vec{\xi}_N)}{\partial \xi_1} & \cdots & \frac{\partial \Psi_P(\vec{\xi}_N)}{\partial \xi_1} \\
 \vdots & \vdots & \ddots & \vdots \\
 \frac{\partial \Psi_0(\vec{\xi}_N)}{\partial \xi_n} & \frac{\partial \Psi_1(\vec{\xi}_N)}{\partial \xi_n} & \cdots & \frac{\partial \Psi_P(\vec{\xi}_N)}{\partial \xi_n}
 \end{bmatrix}
 \begin{pmatrix}
 \alpha_0 \\
 \alpha_1 \\
 \alpha_2 \\
 \vdots \\
 \alpha_P
 \end{pmatrix}
 =
 \begin{pmatrix}
 \alpha^*(\vec{\xi}_0) \\
 \frac{\partial \alpha^*(\vec{\xi}_0)}{\partial Z_1} \frac{\partial Z_1}{\partial \xi_1} \\
 \vdots \\
 \frac{\partial \alpha^*(\vec{\xi}_0)}{\partial Z_n} \frac{\partial Z_n}{\partial \xi_n} \\
 \alpha^*(\vec{\xi}_1) \\
 \frac{\partial \alpha^*(\vec{\xi}_1)}{\partial Z_1} \frac{\partial Z_1}{\partial \xi_1} \\
 \vdots \\
 \frac{\partial \alpha^*(\vec{\xi}_1)}{\partial Z_n} \frac{\partial Z_n}{\partial \xi_n} \\
 \vdots \\
 \vdots \\
 \alpha^*(\vec{\xi}_N) \\
 \frac{\partial \alpha^*(\vec{\xi}_N)}{\partial Z_1} \frac{\partial Z_1}{\partial \xi_1} \\
 \vdots \\
 \frac{\partial \alpha^*(\vec{\xi}_N)}{\partial Z_n} \frac{\partial Z_n}{\partial \xi_n}
 \end{pmatrix}
 \quad (4.7)$$

$$\alpha_i(x, y, z, t) = \left[\Psi_P(\vec{\xi}) \right]^{-1} \alpha^*(x, y, z, t; \vec{\xi})$$

where $N = \frac{P+1}{n+1}$. Note that $\frac{P+1}{n+1}$ is chosen such that N is always an integer value. Furthermore, $\left[\Psi_P(\vec{\xi}) \right]$ is a squared matrix when the number of deterministic solution N_d is equal to N . This method reduces the deterministic black-box evaluations by a factor of $(n + 1)$. In addition, the collocation points are selected based on the OCM method described in Section 4.2.2. Note the stochastic linear system given by Equations 4.7 is inherently unstable due to the presence of sensitivity derivatives. When $N_d > N$, regression analysis is applied to Equation 4.7 because it provides a robust means of estimating the PC coefficients. This is because the influence of each collocation point is moderated by all other collocation points.

4.3 The Application of NIPC methods to a Supersonic Wedge and Expansion Corner

To test the performance of the NIPC methods, the two classical fluid dynamic problems presented in Chapter 3 are repeated. The problems presented here are an inviscid, steady, two-dimensional supersonic flow of a calorically perfect gas over a wedge and an expansion corner. The uncertainty considered herein arises due to uncertainty in free-stream AoA . The free-stream AoA uncertainty was introduced through an angle, α , described by a Gaussian PDF.

In these two test cases, the focus is on the Hermite PC, and its use in propagating uncertainty using the Collocation Method coupled with sensitivity derivatives denoted as the gradient-based NIPC(GBNIPC) method. The term $\frac{\partial \alpha^*(x,y,z,t;\vec{\xi})}{\partial z_i}$ (*e.g.*, see Equation 4.7) used in the GBNIPC method is computed by the complex *FUN3D* flow solver at NASA Langley Research Center. The complex *FUN3D* code is a tetrahedral node-centered upwind solver which has the capability of computing flow field variables sensitivity derivatives in the computational domain by using the Complex Variable formulation described in Section 2.2.1. Note the output sensitivity derivatives of the flow field variables are second order accurate.

In these two test cases, the GBNIPC method is compared to the statistics obtained with MC simulations described in Walters[46]. These MC simulations enable an assessment of the accuracy of the GBNIPC method for the selected problems. Note that performing a large number of MC simulations would be beyond the reach of current computational power, even with high performance computers, for many realistic aerodynamic problems that required the solution of complex flow fields.

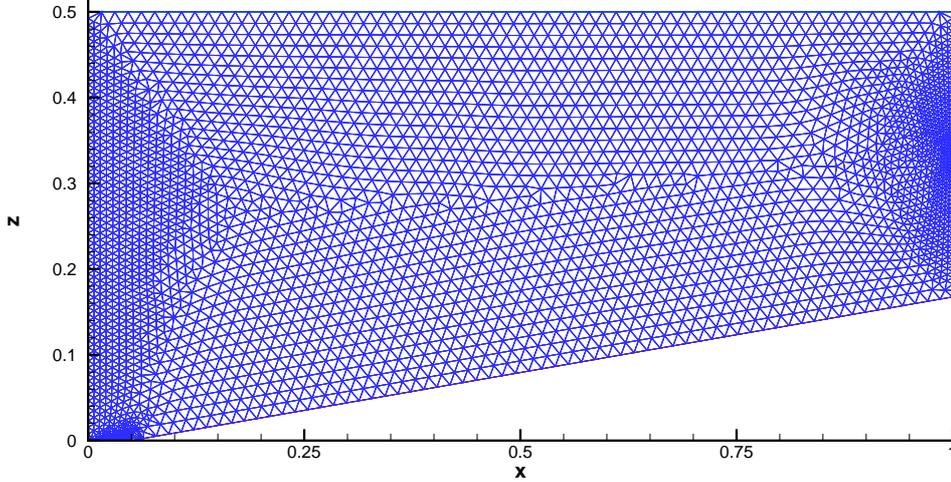


Figure 4.1: An unstructured grid with 3265 *nodes*. The wedge angle is 10° . The Wedge L.E. is 0.05 units

4.3.1 Stochastic Oblique Shock Wave Results

Uncertainty was introduced through the free-stream AoA by,

$$AoA(\xi) = \overline{AoA} + \xi$$

where the mean \overline{AoA} is 0° , and ξ is a normally distributed random variable ($\xi = N[0, 1]$). The effect of free-stream uncertainty AoA is modeled and propagated using the GBNIPC and the MC methods. The deterministic complex $FUN3D$ code is called as a black box by the GBNIPC and MC methods.

The deterministic problem was solved numerically using the complex $FUN3D$ code. The computations were performed for an inviscid, compressible flow on a unstructured wedge grid shown in Figure 4.1. Note the inviscid fluxes on the node cell-faces were calculated using Roe flux difference splitting. The boundary conditions for this problem were set as follows: all flow variables were kept fixed at their free-stream values at the inflow boundary. The free-stream Mach number was chosen

as $M_\infty = 3.0$. For the far-field (top) and outflow surfaces, all flow variables were determined by a first-order extrapolation from the interior cells. An inviscid boundary condition was prescribed along the bottom surface.

In the MC method, 10^3 simulations were generated using 10^3 samples from the $AoA(\xi)$ PDF. For this case, it takes approximately 8.8 hours (*i.e.*, 32 seconds per MC run to drive the residual to machine zero) to compute 1,000 MC solutions, and was sufficient enough to capture the statistics of the output distribution. In the GBNIPC method, a 5^{th} -order PC expansion was chosen to model the uncertainty in the free-stream AoA . The input random variable, $AoA(\xi)$, was sampled from a normal PDF at points $\{-3.32, -1.88, -0.616, 0.616, 1.88, 3.32\}$. These sample points corresponds to the roots of a 6^{th} -order Hermite polynomial. Recall that for the GBNIPC method, the number of sample points are reduced by a factor of $(n + 1)$ or 2 for this case. Thus, the collocation points $\xi_{CP} = \{-0.616, 0, 0.616\}$ were selected, and deterministic solutions were obtained at these points. Note the selected collocation points includes zero although the roots of a 6^{th} -order Hermite polynomial does not contain this root. This is because the OCM method requires the selection of the zero collocation point as described in Section 4.2.2.

Contours of the mean and standard deviation of pressure (P/P_{ref}) obtained via the GBNIPC and MC methods are illustrated in Figure 4.2. Both methods are in excellent agreement. This observation confirms the convergence of the GBNIPC method for a 5^{th} -order PC expansion. To quantify the convergence of the mean and standard deviation of the GBNIPC method, one point was chosen at the exit plane on the wall ($x = 1.0, z = 0.296394$) downstream of the shock depicted in Figure 4.2. At this point, the GBNIPC and the MC estimation of the mean and standard deviation and their associated confidence intervals are presented in Table 4.1. As shown, the GBNIPC estimation of the mean and standard deviation fall within the 95% CI. Although not shown here, a first-order was sufficient to estimate the statistics at the

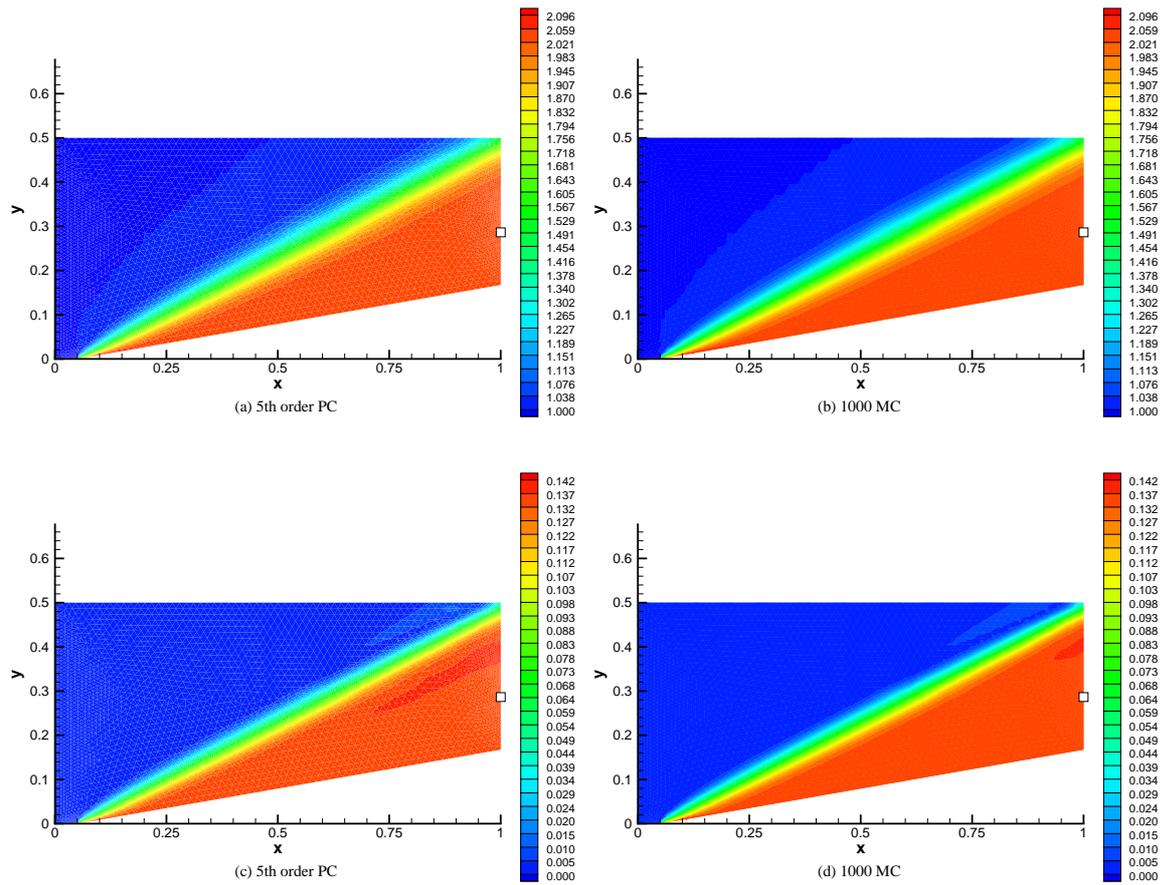


Figure 4.2: Comparison of the mean and standard deviation of (P/P_{ref}) between GBNIPC and MC Methods. (a) Mean P/P_{ref} obtained via GBNIPC method. (b) Mean P/P_{ref} obtained via MC method. (c) Standard Deviation of (P/P_{ref}) obtained via GBNIPC method. (d) Standard Deviation of (P/P_{ref}) obtained via MC method.

Table 4.1: The mean and standard deviation of (P/P_{ref}) obtained with GBNIPC and MC methods at location $(x = 1.0, z = 0.296394)$ for the wedge problem. The 95% confidence intervals for the standard deviations(StD) are calculated from the MC simulations using the Bootstrap method.

	GBNIPC(1 st order)	1000 MC	95% CI
Mean	2.05249	2.05599	[2.04764, 2.0644]
StD	0.135977	0.135514	[0.12982, 0.14108]

exit point. Note the 95% CI was constructed via the bootstrap method using the 1000 MC simulations. Here, 500 bootstrap samples were run for this case.

4.3.2 Stochastic Expansion Wave Results

As in the stochastic oblique shock problem, uncertainty was introduced through the free-stream AoA described by,

$$AoA(\xi) = \overline{AoA} + \xi$$

where the mean \overline{AoA} is 0° , and ξ is a normally distributed random variable($\xi = N[0, 1]$). The effect of free-stream uncertainty $AoA(\xi)$ is modeled and propagated using the GBNIPC and the MC methods. The deterministic complex $FUN3D$ code is called as a black box by the GBNIPC and MC methods.

The deterministic problem was solved numerically using the complex $FUN3D$ code. The computations were performed for an inviscid, compressible flow on a unstructured expansion grid shown in Figure 4.3. Note the inviscid fluxes on the node cell-faces were calculated using Roe flux difference splitting. The same boundary conditions described for the stochastic oblique shock problem are use in this case to find the steady-state solutions. The free-stream Mach number was chosen as $M_\infty = 3.0$.

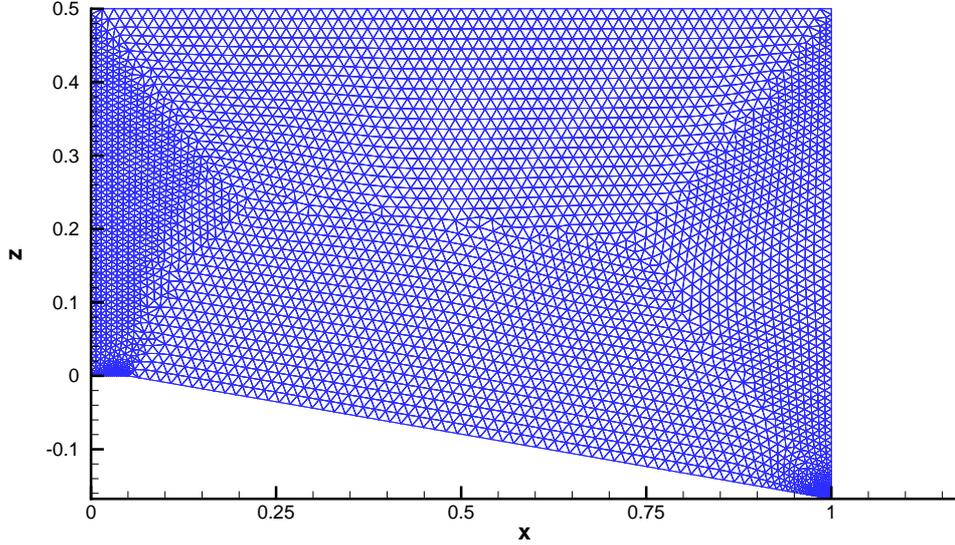


Figure 4.3: An unstructured grid with 4110 *nodes*. The deflection angle is 10° . The Expansion L.E. is 0.05 units

In the MC method, 10^3 solutions were generated using 10^3 samples from the $AoA(\xi)$ PDF. For this case, it takes approximately 9.4 hours (*i.e.*, 34 seconds per MC run to drive the residual to machine zero) to compute 1,000 MC solutions, and was sufficient enough to capture the statistics of the output distribution. In the GBNIPC method, a 5^{th} -order PC expansion was chosen to model the uncertainty in the free-stream AoA . The input random variable, $AoA(\xi)$, was sampled from a Gaussian PDF, and deterministic solutions were obtained at the collocation points $\xi_{CP} = \{-0.616, 0, 0.616\}$.

Contours of the mean and standard deviation of pressure ($P/Pref$) between GBNIPC and MC methods are illustrated in Figure 4.4. Both methods are in excellent agreement. This observation confirms the convergence of the GBNIPC method for a 5^{th} -order PC. Across the expansion wave, all cases show a smooth pressure drop. As in the oblique shock problem, one location ($x = 1.0, z = 0.244897$) is chosen in the

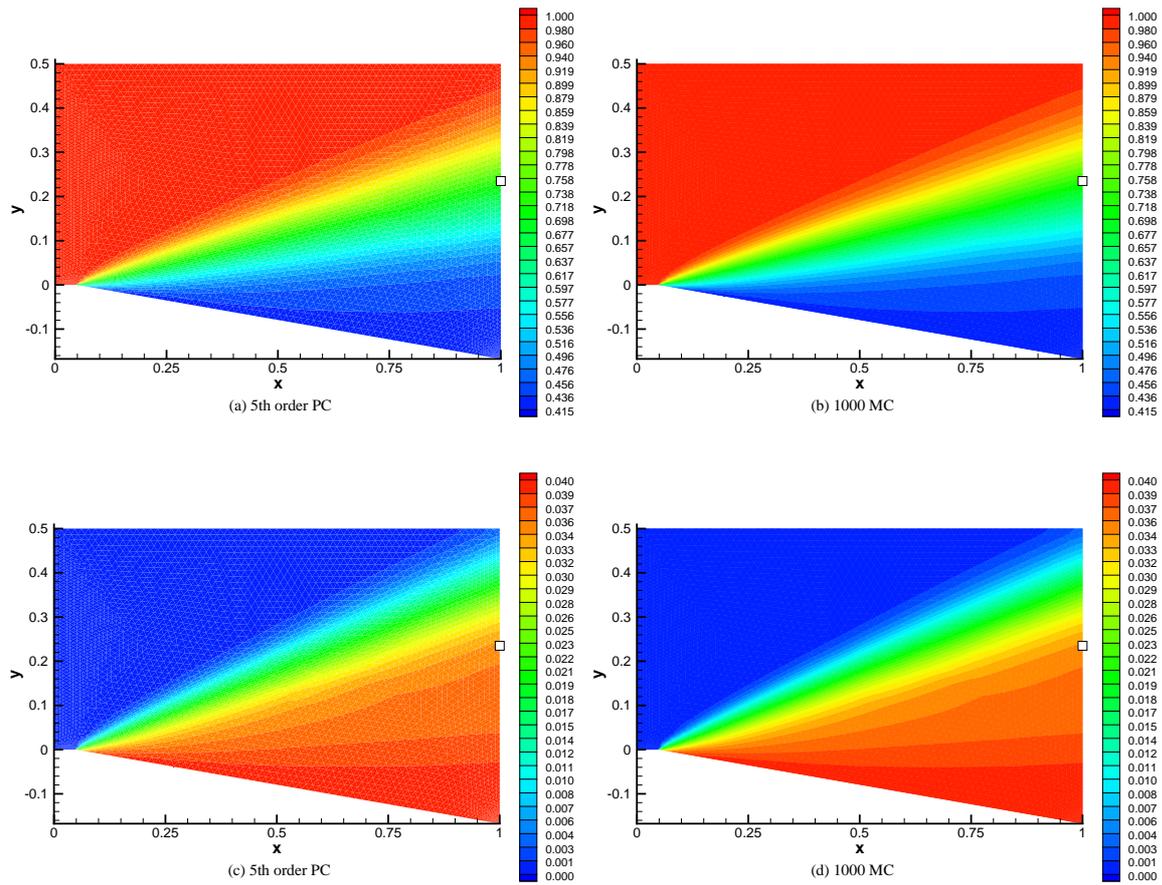


Figure 4.4: Comparison of the mean and standard deviation of (P/P_{ref}) between GBNIPC and MC Methods. (a) Mean P/P_{ref} obtained via GBNIPC method. (b) Mean P/P_{ref} obtained via MC method. (c) Standard Deviation of (P/P_{ref}) obtained via GBNIPC method. (d) Standard Deviation of (P/P_{ref}) obtained via MC method.

Table 4.2: The mean and standard deviation of (P/P_{ref}) obtained with GBNIPC and MC methods at location $(x = 1.0, z = 0.244897)$ for the expansion case. The 95% confidence intervals for the standard deviations(StD) are calculated from the MC simulations using the Bootstrap method.

	GBNIPC(5^{th} order)	1000 MC	95% CI
Mean	0.742916	0.742865	[0.74054, 0.74529]
StD	0.0344393	0.0342788	[0.03263, 0.03570]

flow field to compare the statistics of the GBNIPC and MC methods quantitatively. This location is a point on the wall exit downstream of the expansion fan depicted in Figure 4.4. At this point, the GBNIPC and the MC estimation of the mean and standard deviation and their associated confidence intervals are presented in Table 4.2. As shown, the GBNIPC estimation of the mean and standard deviation fall within the 95% CI. Although not shown here, the GBNIPC estimates do not fall within the 95% CI for an order-chaos lower than 4^{th} -order. This observation is an indication of the additional errors introduced in the GBNIPC method due to the presence of the expansion wave. As a consequence, the expansion region required a high order-chaos to resolve the PC coefficients within the convergence range. Note the 95% CI was constructed via the bootstrap method using the 1000 MC simulations. Here, 500 bootstrap samples were run for this case.

4.4 The Application of NIPC methods to the Onera-M6-Wing Case

The benchmark problem of interest is a compressible, transonic, inviscid flow around the Onera-M6-Wing. The computational domain is bounded by a rectangular box defined by $-6.5 \leq x \leq 6.5$ m, $0 \leq y \leq 4$ m, and $-6.5 \leq z \leq 6.5$ m, in aerodynamic

coordinates relative to a semi-span length of 1 *m*. A typical surface and symmetry plane mesh are shown in Figure 4.5.

The uncertainty considered herein arises due to uncertainty in the free-stream *AoA*, and the free-stream Mach number. The focus is on the Hermite PC, and its use in propagating uncertainty using (1) the Gauss-Hermite method, (2) the Collocation method, and (3) the GBNIPC method described in Sections 4.2.1, 4.2.2, and 4.2.3, respectively.

The output statistics of the NIPC methods are compared to experimental measurements performed by Schmitt and Charpin [72]. In their technical report, they conducted measurements of pressure distributions at several span-wise sections and aerodynamic coefficients of the Onera-M6-Wing for several test conditions. These measurements were obtained in the Onera S2MA wind tunnel at Mach numbers 0.7, 0.84, 0.88, and 0.92 for angles-of-attack from 0° to 6° degrees and a Reynolds number of about 12 million. The level of uncertainty of the pressure coefficient measurements for Mach= 0.84 was reported to be $\Delta C_p = \pm 0.02$.

4.4.1 Grid Convergence Studies for the Onera-M6-Wing Case

Since the level of uncertainty in numerical simulations is dependent upon grid quality, it is essential to verify grid convergence. Furthermore, a lack of grid convergence in a numerical solution contains large discretization errors in the discrete domain of space and time in the governing flow equations and other physical models. In the CFD community, grid convergence is achieved when the numerical solution becomes less sensitive to the grid spacing as the grid is refined.

However, the current practices in grid convergence studies in the field of external aerodynamics are flawed. This observation is evident in the drag prediction workshops (DPWs, [2]). For example, the results of the DPWs participants revealed an

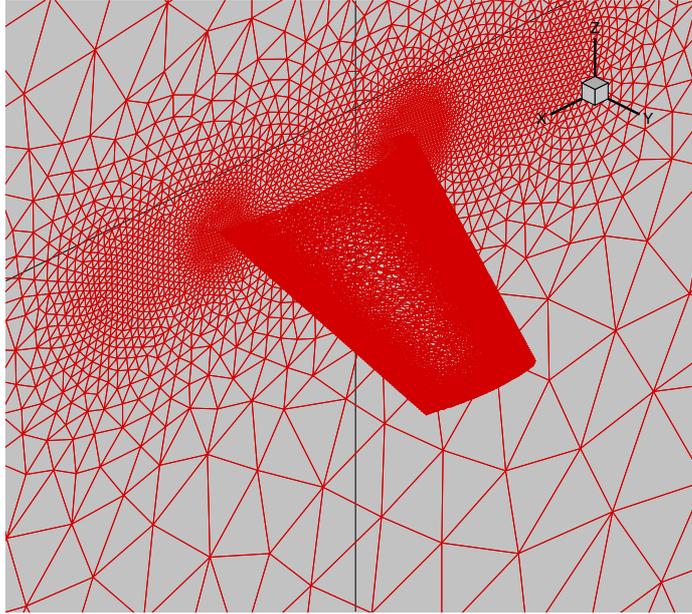


Figure 4.5: A Typical Onera-M6-Wing mesh

apparent lack of grid convergence, which resulted in larger than desired scatter in total drag. The necessary condition to properly establish grid convergence for regular-structured grids was first introduced by Salas [71], and is extended to unstructured grids in this chapter. This necessary condition requires that the aspect grid ratio, χ , be constant over subsequent grid level refinement. Mathematically, this is expressed as

$$\chi = \frac{h_{y,k}}{h_{x,k}} = \frac{h_{z,k}}{h_{x,k}}, \quad \text{for } k = 1, 2, 3, \dots, \quad (4.8)$$

where h is the grid size, k is the mesh sequence level, and x, y, z are the directional coordinates. In the DPWs, the convergence rate, p , is computed for all cases by

$$p = \frac{\ln\left(\frac{\epsilon_{23}}{\epsilon_{12}}\right)}{\ln(r_{12})} \quad (4.9)$$

where,

$$\begin{aligned}\epsilon_{12} &= f_1 - f_2 \\ \epsilon_{23} &= f_2 - f_3 \\ r_{12} &= \left(\frac{N_1}{N_2}\right)^{(1/D)}\end{aligned}$$

and,

f_1 , CFD solution at the fine level

f_2 , CFD solution at the medium level

f_3 , CFD solution at the coarse level

N_1 , number of nodes at the fine level

N_2 , number of nodes at the medium level

D , dimensionality of the problem.

Note that Equation 4.9 is valid for a constant grid aspect ratio. The equation to find the convergence rate p for a grid aspect ratio that is not constant is given by,

$$(r_{12}^p - 1)(r_{23}^p - 1)\epsilon_{12} + (r_{23}^p - 1)\epsilon_{12} - (r_{12}^p - 1)\epsilon_{23} = 0 \quad (4.10)$$

and can be solved for p by implementing Newton's method. Note that Equation 4.10 reduces to Equation 4.9 for a constant grid aspect ratio ($r_{12} = r_{23}$).

4.4.2 Deterministic Onera-M6-Wing Results

The deterministic problem was solved numerically using the complex *FUN3D* code. The boundary conditions for this problem were set as follows: all flow variables were kept fixed at their upstream values at the inflow boundary. The upstream Mach number and *AoA* were set to $Mach = 0.84$ and $\alpha = 3.06^\circ$. For the far-field (top) and outflow surfaces, the flow variables were obtained by the 1-d Riemann invariants. An inviscid boundary condition was prescribed along the surface of the wing. Note

Table 4.3: Grid Level Size

Level	Number of Nodes	Number of Cells
Coarse	34,406	198,367
Medium	110,521	647,447
Fine	235,458	1,390,199
Super-Fine	710,958	4,229,826

the inviscid fluxes on the node cell-faces were calculated using Roe flux difference splitting.

The grids used in the simulations are shown in Table 4.3 and Figure 4.6. These grids were generated using the tetrahedral mesh generation package *VGRID*. Note the grids were sequenced uniformly from coarse to fine using the Element Scaling Software of NASA LaRC.

Figure 4.7 shows the discretization and iterative errors of the total lift and drag coefficients. Note the convergence rate for the C_D is slow to converge, but nevertheless the solutions are monotonic and in the asymptotic region. Using Equation 4.10, the convergence rate is found to be $p = 1.7$ for the lift coefficient, and $p = 0.34$ for the drag coefficient. Note the refinement ratio in Figure 4.7 is define as,

$$(h/h_0)^2 = (N_0/N)^{2/3} \quad (4.11)$$

where N is the number of nodes at a given level, and N_0 is the number of nodes to normalize h . In all deterministic cases, $N_0 = 235,458$. The deterministic pressure contours are shown in Figure 4.8.

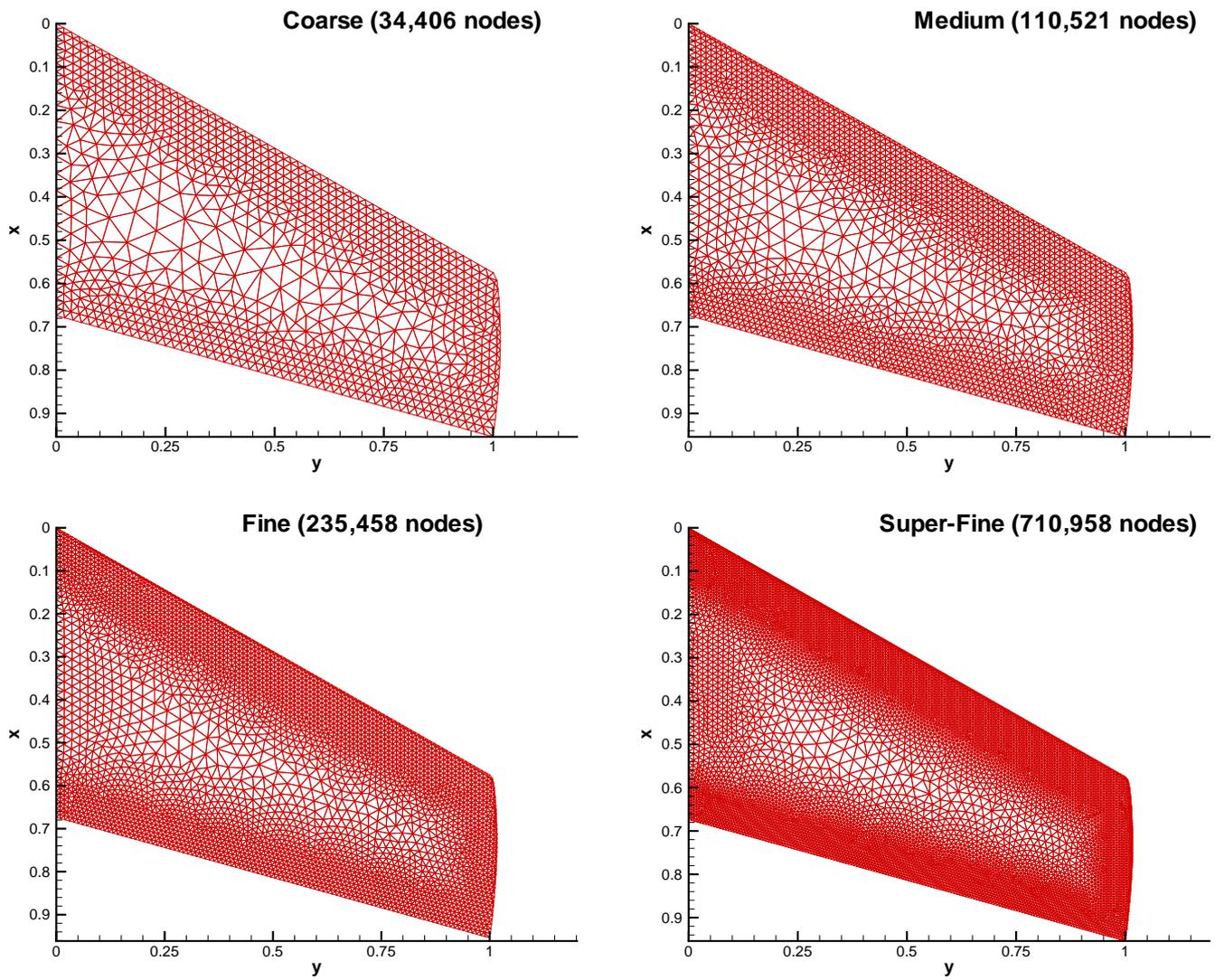


Figure 4.6: Uniform Grid Refinement

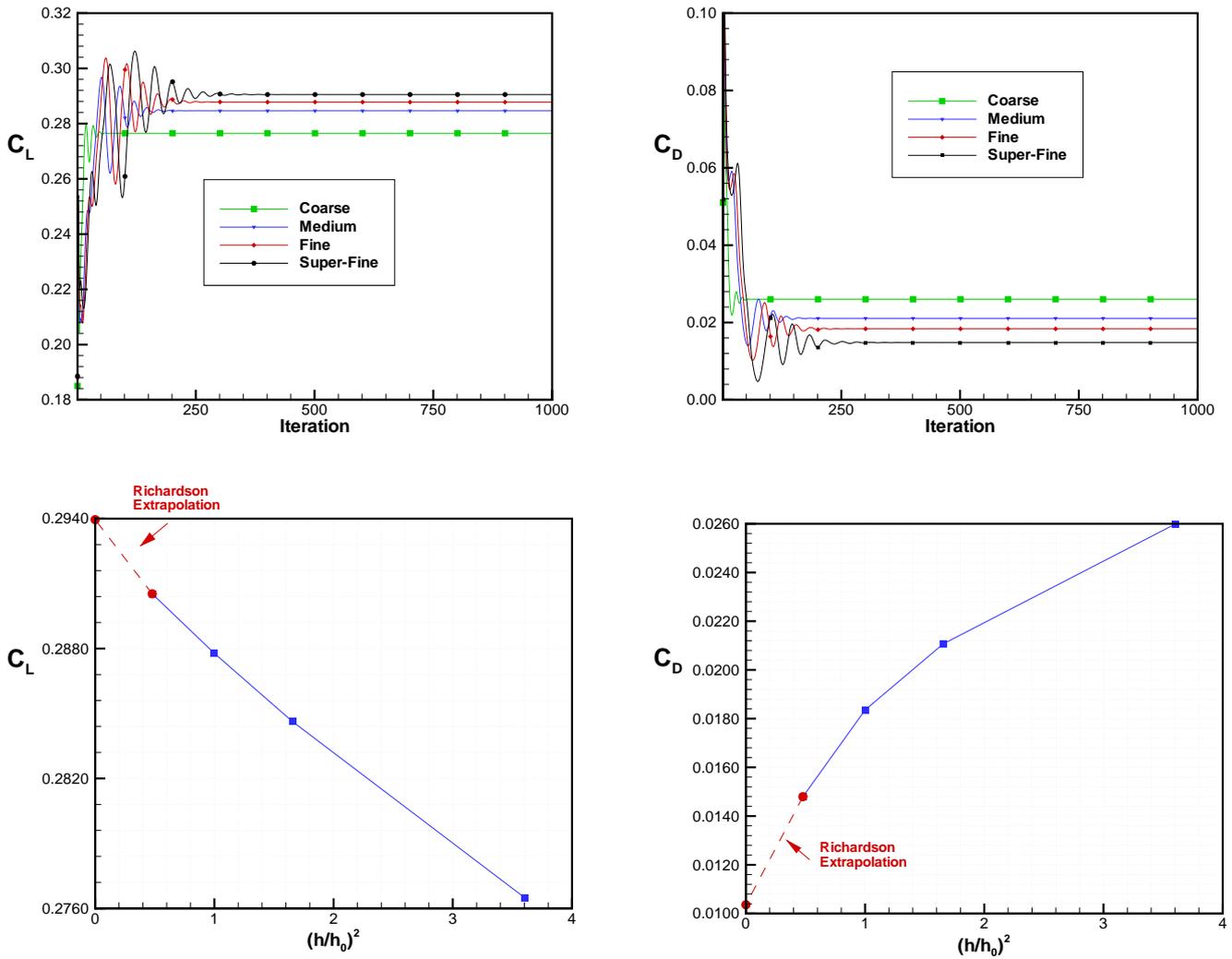


Figure 4.7: The Iterative and Discretization Error

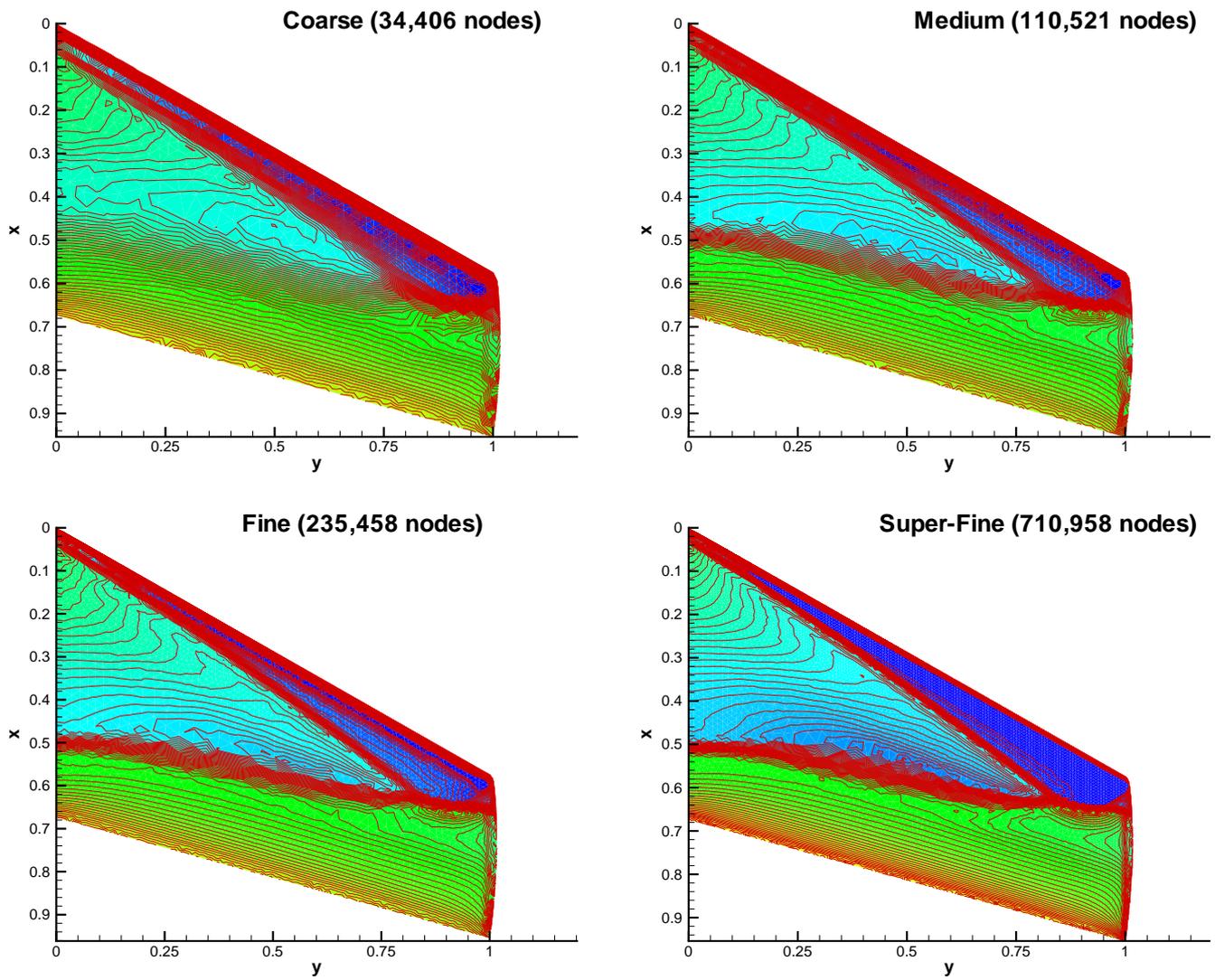


Figure 4.8: Deterministic Pressure Contours

4.4.3 Stochastic Onera-M6-Wing Results

The stochastic Onera-M6-Wing problem was formulated by introducing uncertainty through the AoA and the Mach number(M) given by,

$$AoA(\xi_1) = \overline{AoA} + \sigma_1 \xi_1$$

$$M(\xi_2) = \overline{M} + \sigma_2 \xi_2$$

where ξ_1 and ξ_2 are two normally distributed independent random variables($\xi_1 = N[0, 1]$, $\xi_2 = N[0, 1]$). The mean \overline{AoA} was 3.06° , and the coefficient of variation was 1% (*i.e.*, $\sigma_1 = 0.0306$). The mean Mach number \overline{M} was 0.84, and the coefficient of variation was 0.5% (*i.e.*, $\sigma_2 = 0.0042$). In this application, the focus is on the Hermite PC, and its use in propagating uncertainty using the NIPC methods. The deterministic complex *FUN3D* code is called as a black-box by the NIPC methods. All deterministic runs use the fine grid shown in Figure 4.6, where the CPU time for each deterministic run is approximately 1.5 hours.

Figures 4.9 and 4.10 illustrate the modes of the pressure coefficient for a 4th-order PC at sections $y/b = 0.2$ and $y/b = 0.65$, respectively. For the Gauss-Hermite method, seven integration points were used along each stochastic direction, which required $(2p-1)^n = 49$ deterministic runs for an exact estimation. For the Collocation method, a 4th-order chaos with two random dimensions required $P + 1 = \frac{(n+p)!}{n!p!} = 15$ deterministic runs. For the GBNIPC method, the required number of deterministic runs was reduced to $\frac{2(P+1)}{n+1} = 10$.

The Collocation and GBNIPC methods slightly over-predict the second and third-order modes at the shock location compared to the Gauss-Hermite method at sections $y/b = 0.2$ and $y/b = 0.65$. However, the zeroth and first-order modes obtained from the NIPC methods are in good agreement as shown in Figures 4.9 and 4.10. These observations are an indication of the additional errors introduced in the GBNIPC and the Collocation methods at the shock location downstream of the leading edge. Note

the level of accuracy of the PC coefficients can be increased by running higher order-chaos for the NIPC methods. However, this is not necessary for this case. This is because the modes of the pressure coefficients in regions of high probability(*i.e.*, lower-order-modes) are sufficiently accurate to compute the mean and standard deviation. Note that these statistical parameters(*i.e.*, the mean and standard deviation) are of paramount interest to the aerodynamicist.

Figures 4.11 and 4.12 illustrate the mean pressure coefficient in concert with its 95% CI compared with experimental data at span-wise sections $y/b = 0.2$ and $y/b = 0.65$. Note the 95% confidence intervals were constructed via the bootstrap method using the PC coefficients obtained via the GBNIPC method. Here, 500 bootstrap samples were run. At sections $y/b = 0.2$ and $y/b = 0.65$, the uncertainty band of the pressure coefficient is too narrow to account for the observed discrepancy between prediction and experiment. This indicates that the level of uncertainty in the two input random variables $AoA(\xi)$ and $Mach(\xi)$ do not account for the observed discrepancies between prediction and experiment. Furthermore, the mean pressure coefficient over-predicts the experimental data at the shock locations of the wing as shown in Figures 4.11 and 4.12. The failure of the CFD model to capture the shock locations accurately, also indicates that the observed discrepancies are likely due to the CFD model uncertainty.

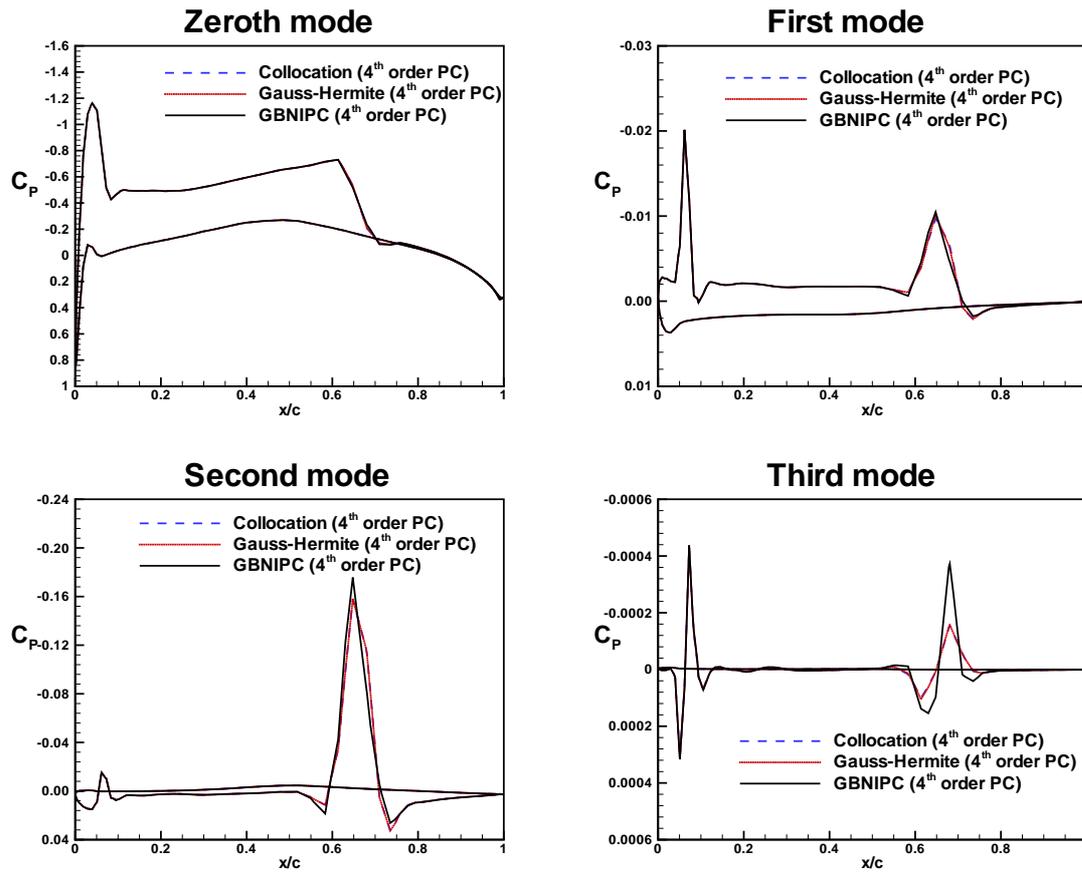


Figure 4.9: The PC modes of the Pressure Coefficient (C_p) at Station $y/b = 0.2$, for input random variables, $\overline{AoA} = 3.06$, $CoV = 1\%$ and $\overline{M} = 0.84$, $CoV = 0.5\%$

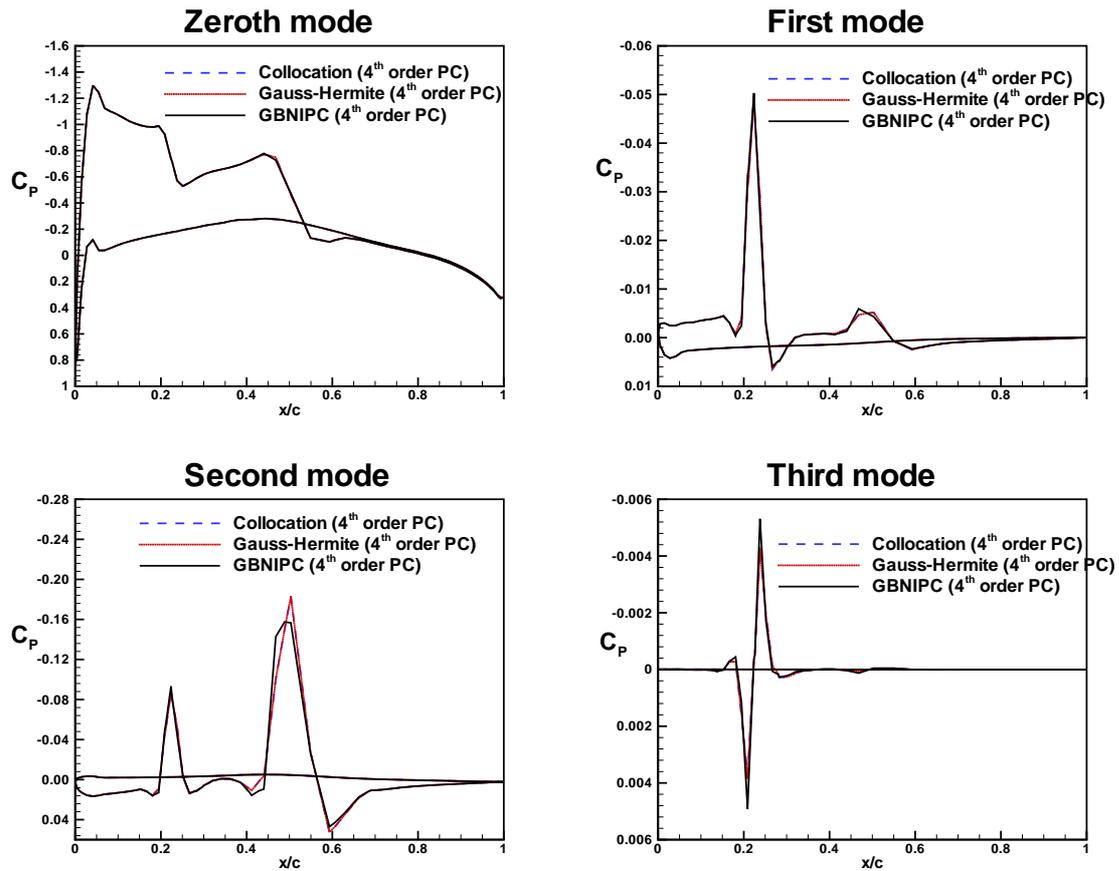


Figure 4.10: The PC modes of the Pressure Coefficient (C_p) at Station $y/b = 0.65$, for input random variables, $\overline{AoA} = 3.06$, $CoV = 1\%$ and $\overline{M} = 0.84$, $CoV = 0.5\%$

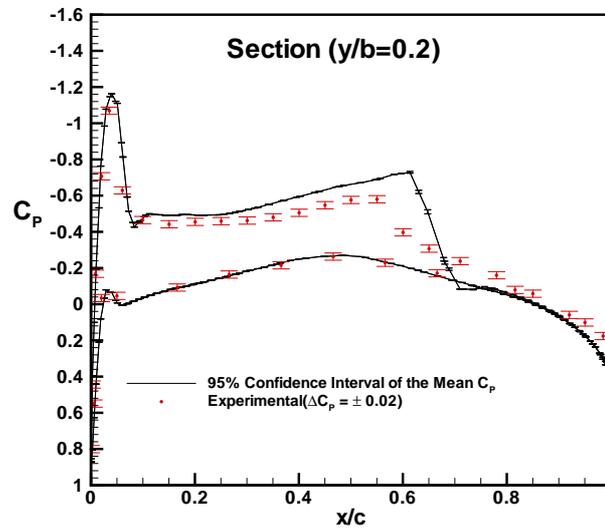


Figure 4.11: 95% CI of the Mean Pressure Coefficient(C_p) compare with Experimental results at Station $y/b = 0.2$, for input random variables, $\overline{AoA} = 3.06$, $CoV = 1\%$ and $\overline{M} = 0.84$, $CoV = 0.5\%$

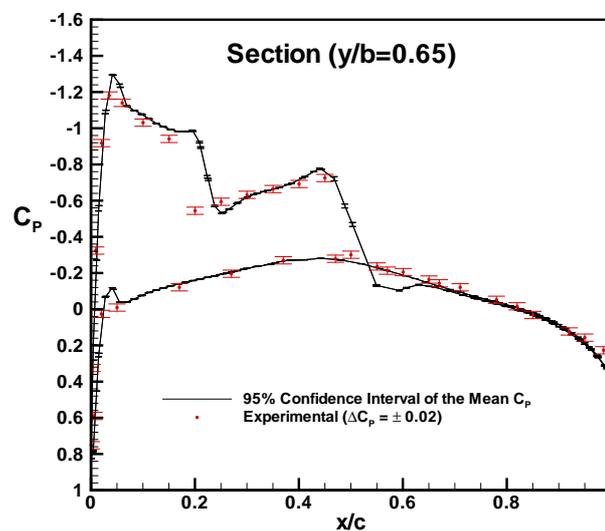


Figure 4.12: 95% CI of the Mean Pressure Coefficient(C_p) compare with Experimental results at Station $y/b = 0.65$, for input random variables, $\overline{AoA} = 3.06$, $CoV = 1\%$ and $\overline{M} = 0.84$, $CoV = 0.5\%$

Chapter 5

Turbulence Model Uncertainty Analysis

5.1 Introduction and Background

The uncertainty of turbulence modeling is manifested in the large variety of available models in the literature. These range from Prandtl's zero-equation mixing length model [40] to more complex detailed turbulence models. Uncertainty in the application of turbulence models arises not only due to uncertainty in model inputs or parameters, but also due to uncertainty in model formulation. As discussed in Chapter 1, uncertainty in model formulation arises for several reasons: (1) Alternative sets of scientific assumptions of a phenomena under study exists, (2) the representation of a phenomena under study is simplified for purposes of mathematical tractability, (3) the inappropriate selection of spatial and temporal resolution in the application of numerical methods, and (4) the inappropriate selection of model boundaries in the application of numerical methods. Examples of sources of model uncertainty in CFD models are presented in Table 1.1.

The limits of applicability of turbulence models can readily be accessed by performing uncertainty analysis. This information not only gives an assessment to the importance and contribution of each source of uncertainty, but also an indication to where the available computational resources should be focused. Moreover, performing uncertainty analysis identifies the robustness of the model assumptions, formulation, and its parameters.

A widely known approach for performing turbulence model uncertainty analysis is to run a number of simulations with a variety of turbulence models, and assess how the modeling effects the results. The procedure described in Isukapalli [47] is typically followed: (1) Evaluate available alternative models ranging from simplified models to more complex detailed models, (2) If the results of low detailed models agree closely with the those of high detailed models, the low detailed models are preferable because they generally required fewer computational resources, and (3) construct uncertainty bounds based on results from different model formulation. There has also been some work perform on turbulence model uncertainty. For example, Boggaard[48] and Godfrey[31] used the SEM to rank the relative contributions of the closure coefficients of turbulence models to the output for a variety of applications.

However, the main drawback of these methods is that the uncertainty in model formulation of one specific model is still not estimated. Furthermore, this approach cannot quantify or identify the sources of model uncertainty that are causing the CFD simulation to differ from their true or exact value. Because of these main limitations, the development of alternative methods is presented in this chapter.

The relevant background information on turbulence modeling is presented. This is followed by a description of the one-equation Spalart-Allmaras(S-A) turbulence model [74]. A methodology for quantifying model uncertainty for the S-A turbulence model is developed, and demonstrated on the Onera-M6-Wing case.

5.2 Background Review on Turbulence Modeling

In practice, most flows of engineering interest are turbulent. For example, flow past rockets, airplanes, automobiles, and internal combustion engines occur well above the critical Reynolds number at which laminar flow exists. Moreover, high Reynolds number turbulent flow involved a wide range of fluctuating velocity scales or eddies that must be resolved. These scales range from the smallest eddies to the largest eddies which are also known as the microstructure and macrostructure scales(see Wilcox [90]), respectively. Since the microstructure scales are much larger than the molecular dimensions, turbulence is a continuum phenomenon. On the other extreme, the macrostructure scales have the largest dimensions, and are comparable to the solid body about which the flow is being computed. Note that the turbulence phenomenon can be thought as a spectrum of eddies of all sizes in between the two extremes. Furthermore, this spectrum increases rapidly as the Reynolds number increases.

To compute an accurate simulation of a turbulent flow, the entire spectrum of active scales must be captured. Since turbulence is a continuum, consequently, the Navier-Stokes, energy and mass-conservation equations captures all of the physics of turbulent fluid motion. However, computing the Navier-Stokes equations directly, exact to the smallest scales, requires too much computational efforts even with current state-of-art computing power. Since the computational demands for solving the exact turbulent-flow equations are so intensive, the Reynolds-Average-Navier-Stokes(RANS) equations are typically implemented in order to reduce computational efforts. Note that the RANS equations introduces additional terms that need to be modeled in order to achieve a closure for the unknowns. Information in the small-scale turbulent fluctuation(i.e., *Kolmogorov*-scales,see Wilcox [91]) is lost due to the averaging procedure in concert with the closure problem implemented in the RANS equations. The RANS equations are closed by making choices for empirical constants,

and incorporating dimensional analysis of the exact behavior of the turbulence quantities as the solid wall of a rigid body is approached(see Hinze [40]).

In the RANS modeling approach, time or mass averaging is implemented in the mass-conservation, momentum, and energy equations to account for turbulent fluctuations. The dependent variables of these equations are split into a mean and a fluctuating component known as the Reynolds decomposition. For example, the Reynolds decomposition for the components of velocity is:

$$u_i(x, y, z, t) = \bar{u}_i(x, y, z) + u'_i(x, y, z, t) \quad \text{with} \quad i = 1, 2, 3$$

For other quantities like density, pressure, and energy, the Reynolds decomposition is:

$$\phi(x, y, z, t) = \bar{\phi}(x, y, z) + \phi'(x, y, z, t).$$

Upon substituting the Reynolds decomposition in the continuity, momentum, and energy equations and using the summation convention, the resulting RANS equations are written as follows:

$$\begin{aligned} \frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_i} (\bar{\rho} \tilde{u}_i) &= 0 \\ \frac{\partial \bar{\rho} \tilde{u}_i}{\partial t} + \frac{\partial}{\partial x_j} (\bar{\rho} \tilde{u}_i \tilde{u}_j + \bar{p} \delta_{ij}) &= \frac{\partial}{\partial x_j} (\tilde{\tau}_{ij}) + \frac{\partial}{\partial x_j} \left(-\bar{\rho} \widetilde{u''_i u''_j} \right) \\ \frac{\partial \bar{\rho} \tilde{e}_0}{\partial t} + \frac{\partial}{\partial x_i} \left(\bar{\rho} \tilde{e}_0 \tilde{u}_i + \bar{p} u_i + \bar{\rho} \widetilde{e''_0 u''_i} \right) &= \frac{\partial}{\partial x_i} (\overline{\tau_{ij} u_j}) - \frac{\partial \bar{q}_i}{\partial x_i} \end{aligned} \quad (5.1)$$

where

$$\begin{aligned} \tilde{e}_0 &= \bar{C}_v \tilde{T} + 1/2 \tilde{u}_i \tilde{u}_i + 1/2 \widetilde{u''_i u''_i} \\ \bar{p} &= \bar{\rho} R \tilde{T} \end{aligned}$$

δ_{ij} is the delta function, the bar($\bar{\cdot}$) denotes time average, and the tilde($\tilde{\cdot}$) denotes mass or Favre[55] average. The additional terms that appears in Equation 5.1(i.e., $-\bar{\rho} \widetilde{u''_i u''_j}$) are called the Reynolds stresses and represent the effects of turbulent fluctuations. Furthermore, these stresses must be modeled in order to close the RANS equations.

The closure model employs the Boussinesq hypothesis[40] for all turbulence models to relate the Reynolds stresses to the mean velocity gradients given by,

$$-\overline{\rho u_i'' u_j''} = \mu_t \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial \tilde{u}_k}{\partial x_k} \right) \quad (5.2)$$

where μ_t is the turbulent eddy viscosity. According to the Boussinesq hypothesis, the Reynolds stresses are assumed to behave in analogy to the laminar viscous diffusion terms, and the turbulent eddy viscosity is assumed to be an isotropic scalar quantity(i.e., the eddy viscosity is invariant with respect to direction).

These assumptions, however, are not strictly true, and consequently give rise to turbulence model uncertainty. Uncertainty in turbulence model formulation, which assumes the Boussinesq hypothesis, are addressed in the context of the S-A turbulence model. This is the only one-equation turbulence model that will be treated, and is discussed in the next section.

5.2.1 S-A Turbulence Model

The S-A turbulence model [74] is based on a postulated transport equation comprised of terms modeling convection, diffusion, production, and dissipation of turbulence eddy viscosity. It is a relatively simple model that has been optimized for aerodynamic applications, most notably for flow past a wing. One of the main advantage of the S-A model is the simplicity in imposing the free-stream and wall boundary conditions. In a near wall-region, this model depends on the distance to the closest wall d to reproduce the viscous effects in the laminar-sublayer. Far from the wall, the viscous effects becomes negligible. The transport equation for the modified turbulent viscosity $\tilde{\nu}$ can be written using the summation convention as follows,

$$\frac{D(\rho\tilde{\nu})}{Dt} = \underbrace{\rho C_{b1} \tilde{S} \tilde{\nu}}_{Production} + \underbrace{\frac{\partial}{\partial x_j} \left[\frac{(\mu + \rho\tilde{\nu})}{\sigma} \frac{\partial \tilde{\nu}}{\partial x_j} \right]}_{Diffusion} + \underbrace{\frac{\rho C_{b2}}{\sigma} \left(\frac{\partial \tilde{\nu}}{\partial x_j} \right)^2 - \rho C_{w1} f_w \left(\frac{\tilde{\nu}}{d} \right)^2}_{Dissipation} \quad (5.3)$$

Table 5.1: Constants for the S-A model

C_{v1}	7.1
C_{b1}	0.1355
C_{b2}	0.622
C_{w1}	$C_{b1}/\kappa^2 + (1 + C_{b2})/\sigma$
C_{w2}	0.3
C_{w3}	2.0
σ	0.66667
κ	0.41

where,

$$\begin{aligned}\tilde{S} &\equiv S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}, \quad \chi = \frac{\tilde{\nu}}{\nu} \\ S &= \sqrt{2\Omega_{ij}\Omega_{ij}}, \quad \Omega_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right) \\ f_w &= g \left[\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6} \right]^{1/6}, \quad g = r + C_{w2} (r^6 - r), \quad r \equiv \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}\end{aligned}$$

and the turbulent eddy viscosity is $\mu_t = \rho\tilde{\nu}f_{v1}$. The term on the left hand side of Equation 5.3 represents the rate of increase and convection of turbulent viscosity. The first term on the right hand side of Equation 5.3 represents the production of turbulent viscosity. The diffusion term represents the transport of both molecular and turbulent viscosity. The last term represents the dissipation of turbulent viscosity in a near-wall-region due to wall blocking and viscous damping. Far from the wall the dissipation term becomes negligible. The closure-coefficients for the S-A turbulence model are given in Table 5.1.

The S-A turbulence model is coupled with a wall-function that resolves the inner portion of the turbulent boundary layer. In this approach, the numerical solution of the S-A model in the outer region is matched with a wall-function solution of the inner region. The selected wall-function is a law-of-the-wall expression [86] derived

by Spalding in 1961. With a single function, it models the laminar sublayer, a buffer-layer, and the logarithmic layer of the turbulent boundary layer:

$$n^+ = u^+ + e^{-\kappa B} \left[e^{\kappa u^+} - 1 - \kappa u^+ - \frac{(\kappa u^+)^2}{2} - \frac{(\kappa u^+)^3}{6} \right] \quad (5.4)$$

where $\kappa = 0.41$ and $B = 5.5$ for smooth walls. The dimensionless distance to the wall n^+ and dimensionless tangential velocity u^+ are given by,

$$n^+ = \frac{\rho_w d u_*}{\mu_w}$$

$$u^+ = \frac{|\vec{V}_d|}{u_*}$$

Here ρ_w, μ_w are the fluid density and laminar viscosity on the surface, respectively, $|\vec{V}_d|$ is the velocity magnitude at an adjacent point located a normal distance d to the wall, and u_* is the friction velocity.

The S-A turbulence model is derived using (1) selective molecular viscosity dependence, (2) dimensional-analysis, (3) empiricism, and (4) Galilean-invariance of the turbulence viscosity[74]. Even for this high level of turbulence modeling, a potential deficiency in the modeling process or uncertainty in the S-A model still exists. Furthermore, the level of accuracy or uncertainty in the model predictions is highly dependent on the type of application. As a consequence, the S-A turbulence model is validated by numerous comparisons with measurements for carefully documented flows.

The level of accuracy in the predictions of any turbulence model can be improved by (1) model reformulations, and (2) model calibration. Model reformulation involves the discovery and development of a new mathematical formulation that models physical processes or parameter information that was missing in the prior model. For example, Dacles-Mariani[16] proposed a modification of the S-A model which also accounts for the effect of mean strain rate on turbulence production. The modification

is prescribed to the term S in the S-A model given by Equation 5.3 as;

$$S \equiv |\Omega_{ij}| + C_{prod} \min(0, |S_{ij}| - |\Omega_{ij}|)$$

where

$$\begin{aligned} C_{prod} &= 2.0 \\ |\Omega_{ij}| &\equiv \sqrt{\Omega_{ij}\Omega_{ij}} \\ |S_{ij}| &\equiv \sqrt{S_{ij}S_{ij}} \\ \Omega_{ij} &\equiv \frac{1}{2} \left(\frac{\partial u_j}{\partial x_i} - \frac{\partial u_i}{\partial x_j} \right) \\ S_{ij} &\equiv \frac{1}{2} \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right) \end{aligned}$$

Model calibration involves the adjustment of the original model's closure coefficients to new applications. For example, as shown by Wilcox [90], the measured spreading rate and velocity profile can be closely matched by assuming the mixing length $l_m = \alpha\delta$, where α is a closure coefficient and δ is shear-layer thickness. The value of the closure coefficient α are adjusted according to the type of flow as,

$$\alpha = \begin{cases} 0.180, & \text{Far wake} \\ 0.071, & \text{Mixing layer} \\ 0.098, & \text{Plane Jet} \\ 0.080, & \text{Round Jet} \end{cases}$$

In this example, the uncertainty in the closure coefficient α contributes to model uncertainty. This is manifested in the range of values α can assume (e.g., 0.071–0.180) according to the type of application.

5.2.2 Model Uncertainty Analysis

Model uncertainty analysis is accomplished by performing sensitivity analysis coupled with the Collocation method on the incoming turbulence and closure coefficients of

the S-A turbulence model. As discussed in Chapter 2, sensitivity analysis is performed by the complex *FUN3D* flow solver using a complex variable formulation. In this approach, an output CFD variable, f , is expanded in a Taylor series using a complex perturbation parameter h as:

$$f(x + ih) = f(x) + ih \frac{df}{dx} - \frac{h^2}{2} \frac{d^2f}{dx^2} - \frac{ih^3}{6} \frac{d^3f}{dx^3} + \frac{h^4}{24} \frac{d^4f}{dx^4} + \dots \quad (5.5)$$

Solving this equation for the imaginary part of the function yields

$$\frac{df}{dx} \approx \frac{Im[f(x + ih)]}{h} + O(h^2) \quad (5.6)$$

The implementation of this method is straightforward. That is, by declaring all variables of a function as complex and applying complex perturbation to the design variable of interest, the sensitivity derivative of a design variable can be obtained by evaluating the imaginary part of a function. Thus, for example, applying complex perturbations to the incoming turbulence, and evaluating the imaginary part of lift C_L and drag C_D coefficients yields:

$$S_{C_L} \equiv \frac{\partial C_L}{\partial \tilde{v}_\infty} = \frac{Im[C_L(\tilde{v}_\infty + i\Delta)]}{\Delta}$$

$$S_{C_D} \equiv \frac{\partial C_D}{\partial \tilde{v}_\infty} = \frac{Im[C_D(\tilde{v}_\infty + i\Delta)]}{\Delta}$$

where the complex perturbation used in *FUN3D* is $\Delta = 10^{-50}$. Similar sensitivity results can be obtained for other closure coefficients. Note that the complex *FUN3D* code was modified to account for input complex perturbations of the incoming turbulence and the closure coefficients. The Collocation method, discussed in Chapter 4, is then used to propagate the model uncertainty in the incoming turbulence and the closure coefficients. Only the parameters with the strongest influence are included in the analysis which are provided by the sensitivity analysis.

5.3 The Onera-M6-Wing Case

The benchmark problem of interest is a compressible, transonic, fully turbulent flow around the Onera-M6-Wing. On the wing surface at the midchord, n^+ was set equal to 2. The free-stream flow conditions (*i.e.*, the mach number, Reynolds number, angle-of-attack) were set to $M_\infty = 0.8447$, $Re_{\bar{c}} = 11.7 \times 10^6$, and $\alpha = 5.06^\circ$. The computational domain is bounded by a rectangular box defined by $-6.5 \leq x \leq 6.5 m$, $0 \leq y \leq 4 m$, and $-6.5 \leq z \leq 6.5 m$, in aerodynamic coordinates relative to a semi-span length of $1 m$. The viscous surface and symmetry plane mesh was generated from the inviscid fine mesh in Chapter 4 using the package *VGRID*. This viscous mesh is shown in Figure 5.1. On the Onera-M6-Wing, no-slip surface, the turbulence variable $\tilde{\nu}$ is set to zero. For the far-field (top) and outflow surfaces, boundary conditions are imposed by applying the 1-d Riemann invariants to the turbulence variable $\tilde{\nu}$ and the flow-field variables. At the inflow boundary, the Mach number was set to $M_\infty = 0.8447$. Note that the free-stream eddy viscosity value used for the S-A model is $\mu_{t,\infty} = 0.009$. This value can be obtained by setting the turbulence variable to $\tilde{\nu}_\infty = 1.341946$ at the free-stream. For example, the eddy viscosity for the S-A turbulence model is computed at the free-stream by,

$$\begin{aligned} \mu_{t,\infty} &= \rho_\infty \tilde{\nu}_\infty f_{v1} & (5.7) \\ &= \frac{\tilde{\nu}_\infty^4}{\tilde{\nu}_\infty^3 + C_{v1}^3} \\ &\simeq 0.009 \end{aligned}$$

The S-A model uncertainty considered herein arises due to uncertainty in the incoming turbulence $\tilde{\nu}_\infty$, and the closure coefficients C_{b1} , C_{b2} , C_{w2} , C_{w3} , σ , κ . The focus is on the Hermite PC, and its use in propagating uncertainty into the lift and drag coefficients. The output statistics of the Collocation method are compared to experimental measurements performed by Schmitt and Charpin [72]. In their technical report, they conducted measurements of pressure distributions at several spanwise

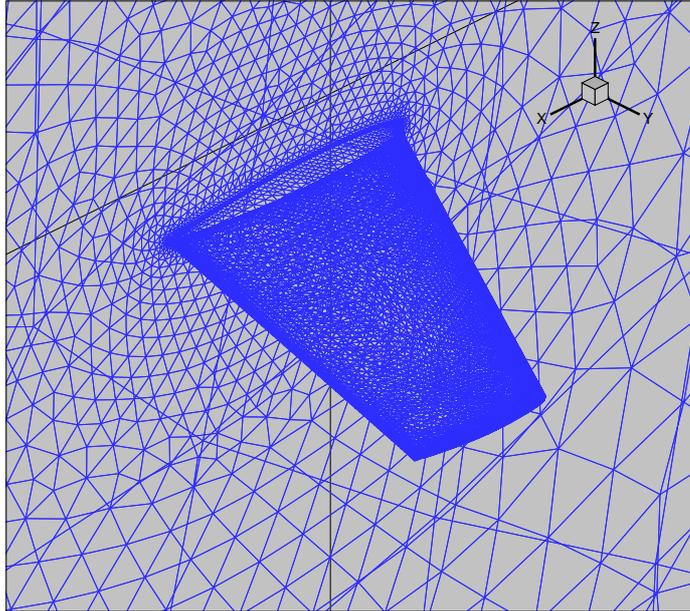


Figure 5.1: Onera-M6-Wing viscous mesh (409,135 nodes)

Table 5.2: Scaled sensitivity of the lift coefficient S_{C_L} and the drag coefficient S_{C_D}

$\times 10^5$	C_{v1}	C_{b1}	C_{b2}	C_{w2}	C_{w3}	σ	κ	\tilde{v}_∞
scaled- S_{C_L}	+177	606	15.0	23.7	+62.0	74.6	-484	-8.49
scaled- S_{C_D}	-16.7	301	3.93	32.2	-4.97	112	+315	+3.79

stations and aerodynamic coefficients of the Onera-M6-Wing for several test conditions. These measurements were obtained in the Onera S2MA wind tunnel at Mach numbers 0.7, 0.84, 0.88, and 0.92 for angles-of-attack from 0° to 6° and a Reynolds number of about 12 million.. The level of uncertainty of the aerodynamic coefficient measurements for Mach= 0.8447 and an angle-of-attack of 5.06° was reported as $C_{L,exp} \approx 0.436 \pm 5$ drag-counts and $C_{D,exp} \approx 0.05 \pm 1.5$ drag-counts.

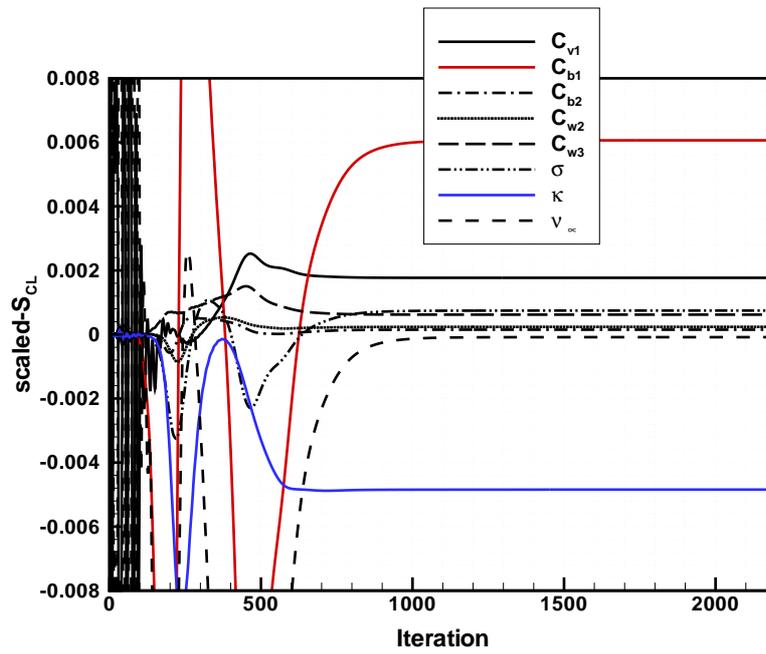


Figure 5.2: Scaled sensitivities of the lift coefficient S_{C_L}

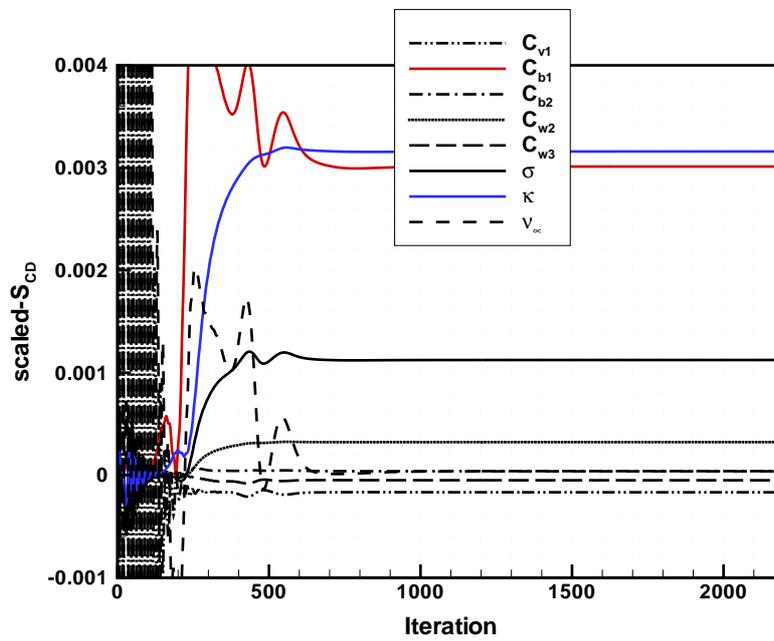


Figure 5.3: Scaled sensitivities of the drag coefficient S_{C_D}

5.3.1 The Onera-M6-Wing Sensitivity results

Figure 5.2 illustrates the scaled sensitivities of the lift coefficient S_{C_L} due to the incoming turbulence $\tilde{\nu}_\infty$ and the closure-coefficients of the S-A turbulence model. For comparison purposes, the scaled sensitivity method, described in Borggaard[14], is implemented. That is, the resulting sensitivities are multiply by the nominal values given in Table 5.1. Note the computation was carried out all the way to 2200 iterations in order to minimize the iterative convergence error. As shown in Figure 5.2, all scaled sensitivities are stabilized after 500 iterations. As presented in Table 5.2, the scaled sensitivities indicate that parameter Cb_1 , which corresponds to the term that models the production of turbulent viscosity, exerts the strongest influence on the lift coefficient in absolute value. Parameters κ and C_{v1} , which correspond to the wall function and eddy viscosity terms, also exert a strong influence on the lift coefficient. In contrast, the parameters C_{b2} , C_{w2} , C_{w3} , and σ , which correspond to the diffusion and dissipation terms exert a relatively weak influence on the lift coefficient. Furthermore, the incoming turbulence $\tilde{\nu}_\infty$ has the least influence on the lift coefficient in absolute value.

Figure 5.3 illustrates the scaled sensitivities of the drag coefficient S_{C_D} due to the incoming turbulence $\tilde{\nu}_\infty$ and the closure coefficients of the S-A turbulence model. As presented in Table 5.2, the scaled sensitivities indicate that parameter κ , which corresponds to the wall function given by Equation 5.4, exerts the strongest influence on the drag coefficient in absolute value. Parameters C_{v1} , C_{b2} , C_{w2} , C_{w3} , and σ , which correspond to the eddy viscosity, diffusion and dissipation terms, exert a relatively weak influence on the drag coefficient. Furthermore, $\tilde{\nu}_\infty$ has the least influence on the drag coefficient in absolute value.

5.3.2 Stochastic Onera-M6-Wing Results

The stochastic Onera-M6-Wing problem was formulated by introducing uncertainty in the S-A turbulence model through the incoming turbulence and closure coefficients. In this application, the focus is on the Hermite PC, and its use in propagating model uncertainty using the Collocation method. In this method, the solution of the lift and drag coefficients are expanded to a 3^d -order Hermite chaos. Although the total number of parameters in the S-A turbulence model is eight for this application, uncertainty is only introduced to the parameters with the strongest influence on the lift and drag coefficient, namely, C_{v1} , C_{b1} , σ , and κ .

Model uncertainty is introduced through uniformly distributed closure coefficients $\overline{C}_{v1}[1 \pm \varepsilon]$, $\overline{C}_{b1}[1 \pm \varepsilon]$, $\overline{\sigma}[1 \pm \varepsilon]$, and $\overline{\kappa}[1 \pm \varepsilon]$. Since Hermite polynomials represent normal random variables, the direct transformation of a uniform $[a, b]$ random variable as function of a normal(0, 1) random variable ξ is required. This is given by:

$$Z(\xi) = a + (b - a) \left\{ \frac{1}{2} + \frac{1}{2} E_{rf}(\xi/\sqrt{2}) \right\}$$

Note that representations of random variables as functions of other random variables are available in the literature [47]. Using the relation above, the direct transformation of the closure coefficients are given by:

$$\begin{aligned} C_{v1}(\xi_1) &= \overline{C}_{v1} \left[1 + \varepsilon \left\{ E_{rf}(\xi_1/\sqrt{2}) \right\} \right] \\ C_{b1}(\xi_2) &= \overline{C}_{b1} \left[1 + \varepsilon \left\{ E_{rf}(\xi_2/\sqrt{2}) \right\} \right] \\ \sigma(\xi_3) &= \overline{\sigma} \left[1 + \varepsilon \left\{ E_{rf}(\xi_3/\sqrt{2}) \right\} \right] \\ \kappa(\xi_4) &= \overline{\kappa} \left[1 + \varepsilon \left\{ E_{rf}(\xi_4/\sqrt{2}) \right\} \right] \end{aligned}$$

where ξ_1 through ξ_4 are normally distributed independent random variables, and the mean of these parameters(i.e., \overline{C}_{v1} , \overline{C}_{b1} , $\overline{\sigma}$, and $\overline{\kappa}$) are given in Table 5.1. As a first

approximation, the error ε in the uncertainty interval of each random variable is set to 5%.

Figures 5.4 and 5.5 illustrate the mean lift and drag coefficient, respectively, in concert with their 95% confidence intervals compare to experimental data. The 95% confidence intervals were constructed via the bootstrap method using the PC coefficients. Here, 500 bootstrap samples were ran. A 3rd order chaos with four random dimensions required 35 deterministic runs for both cases. The deterministic complex *FUN3D* code is called as a black-box, where the CPU time for each deterministic run is approximately 2.0 hours. The uncertainty band of the lift of coefficient is too narrow to account for the observed discrepancy between prediction and experiment. Moreover, the level of uncertainty in the closure coefficients do not account for the observed discrepancies between the lift coefficient of prediction and experiment. As shown in Figure 5.5, the uncertainty band in drag coefficient is wider than the lift coefficient. Furthermore, the level of uncertainty in the closure coefficients seems to have some effect on the drag coefficient. However, the uncertainty band in the drag coefficient does not overlap the experimental uncertainty band.

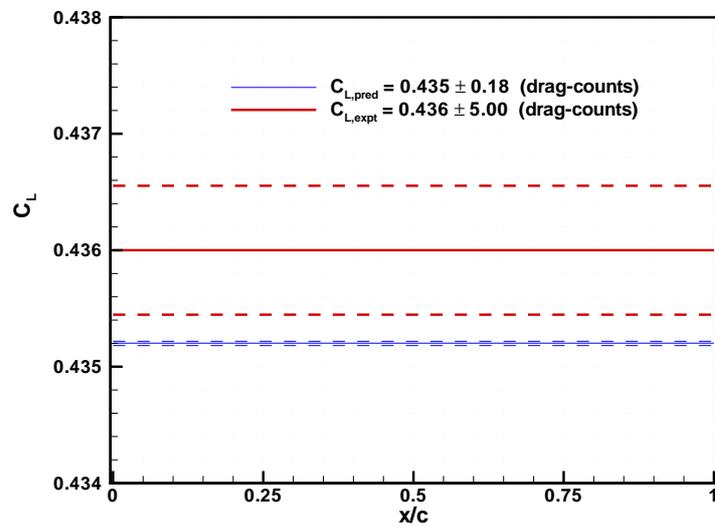


Figure 5.4: Comparison of the lift coefficient C_L between prediction and experiment

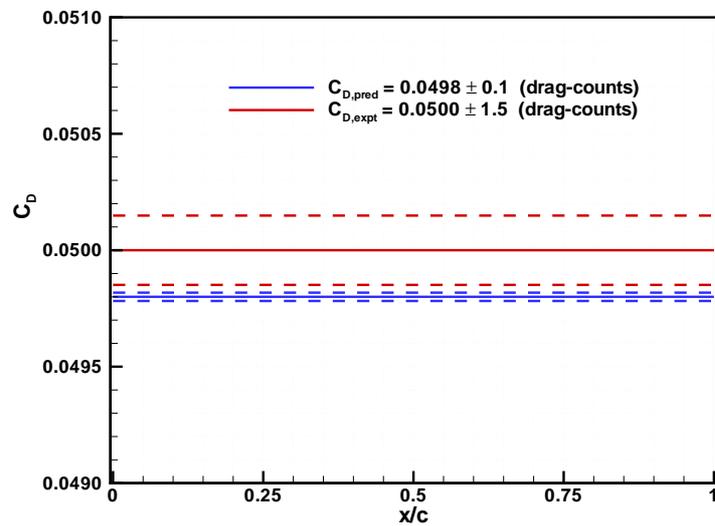


Figure 5.5: Comparison of the drag coefficient C_D between prediction and experiment

Chapter 6

Conclusions and Discussions

The application of CFD models involves significant uncertainties that may have implications on the reliability of its predictions. For example, input parameters in CFD simulations such as geometry, free-stream flow conditions, and angle-of-attack are mostly uncertain, and the variability associated with them can have substantial impact on the final result. Hence, it is important to address these uncertainties. However, the main limitations in performing uncertainty analysis of CFD models using conventional methods are associated with cost and effort. For these reasons, there is a need for the development and implementation of efficient stochastic CFD tools for performing uncertainty analysis.

6.1 Development and Implementation of the Intrusive Method

One of the main contributions of this research is the development and implementation of an implicit formulation for the Euler equations using PC for uncertainty representation and propagation(Chapter 3). In this approach, all dependent variables and random parameters in the Euler equations were replaced with the PC expansions.

The resulting equations were then projected onto the k^{th} basis by using the definition of the inner product. These projected equations resulted in $(P + 1)$ additional deterministic equations, which were solved by the same conventional numerical technique applied to the original deterministic system. Details were presented for the full flux vector and the Van Leer split flux vector as well as their associated Jacobian matrices in a compact PC form.

The Intrusive method has been applied to the two-dimensional Euler equations of gas dynamics for the unit problems: (1) flow over a wedge at supersonic speed, (2) flow over an expansion corner at supersonic speed, and (3) flow over a cosine airfoil at supersonic speed. Comparisons of first-order PC results show good agreement with the MC simulations in which 10,000 realizations were obtained. Although straightforward in theory, an intrusive formulation for complex problems such as the Navier-Stokes simulation of 3-D, viscous, turbulent flows around realistic aerospace vehicles, can be relatively difficult, expensive, and time consuming to implement.

6.2 Development and Implementation of the NIPC Methods

To overcome the drawbacks associated with the intrusive approach, NIPC methods have been developed for uncertainty representation and propagation (Chapter 4). In this new approach, no modification to a deterministic code is required. The deterministic code is called as a black box, and uncertainty is modeled and propagated solely by the PC expansions. This was accomplished by developing and implementing three NIPC methods based on (1) the Galerkin method, (2) the Collocation method, and (3) the GBNIPC method.

In the Galerkin method, the CFD solution is projected onto the PC k^{th} basis by using the definition of the inner product and the property of orthogonality. The

resultant integral of the inner product can be estimated using the Gauss quadrature method. For a problem with n number of random variables, the total number of deterministic solutions required for an exact estimation is $N_d \geq (2p - 1)^n$, where p is the order of the chaos.

In the collocation method, the CFD model random inputs and outputs are approximated by the PC expansions. These expansions contain unknown coefficients of the outputs which are calculated by solving a linear system of equations that uses a selected number of collocation points. The collocation points are selected based on the OCM [79]. For a problem with n number of random variables, the total number of deterministic solutions required is $N_d \geq \frac{(n+p)!}{n!p!}$, where p is the order of the chaos.

In the GBNIPC method, the collocation method is coupled with the Complex Variable sensitivity analysis method in order to improve the computational efficiency. The Complex Variable technique is straightforward to apply and produces accurate sensitivity derivatives without suffering from step size related numerical problems (e.g., see [58]). By declaring all variables of a function as complex and applying complex perturbation to the design variable of interest, the sensitivity derivative of a design variable can be obtained by evaluating the imaginary part of a function. For a problem with n number of random variables, the total number of deterministic solutions required is $N_d \geq \frac{K}{(n+1)}$, where K is the number of PC coefficients. For this method, the number of deterministic solutions should always be greater than required in order to achieve robustness.

The performance of the NIPC methods have been tested on three stochastic fluid dynamic problems: (1) a compressible, supersonic, inviscid flow over a wedge corner with uncertainty in angle-of-attack, (2) a compressible, supersonic, inviscid flow over an expansion corner with uncertainty in angle-of-attack, and (3) a compressible, transonic, inviscid flow around the Onera-M6-Wing with uncertainty in angle-of-attack and Mach-number.

In the oblique shock case, the statistics (i.e., the mean and standard deviation of pressure) obtained with the GBNIPC method were in good agreement with the results of the MC simulations. For example, a first order chaos was sufficient to estimate the statistics at the exit point. This observation implies that the distribution at this point is Gaussian. In the expansion case, a fifth order polynomial expansion is needed to approximate statistics at the exit point. This observation implies that the distribution at this point is Non-Gaussian. For both cases(i.e., shock and expansion cases), the values of the statistics fall within the 95% CI. However, the GBNIPC method required significantly fewer deterministic runs compared to the MC method. For example, the number of deterministic runs used for the shock and expansion cases is $N_d = 2$ and $N_d = 6$, respectively, compared to 1000 MC runs.

In the Onera-M6-Wing case, the necessary condition to properly establish grid convergence is extended to unstructured meshes. This necessary condition, first introduced by Salas[71] for regular-structured grids, requires that the aspect grid ratio χ be constant over subsequent grid level refinement. The modes of the pressure coefficient for a 4th-order chaos was calculated using the NIPC methods. For the Gauss-Hermite method, seven integration points were used along each stochastic direction, which required $(2p - 1)^n = 49$ deterministic runs for an exact estimation. For the Collocation method, a 4th order chaos with two random dimensions required $\frac{(n+p)!}{n!p!} = 15$ deterministic runs. For the GBNIPC method, the required number of deterministic runs was reduced to $\frac{2K}{n+1} = 10$. The CPU time for each deterministic run is approximately 1.5 hours.

The Collocation and GBNIPC methods slightly over-predict the second and third modes at the shock location at various stations compared to the Gauss-Hermite method. This observation suggests that the sensitivity derivatives are introducing small errors in the PC coefficients. The mean pressure coefficient in concert with its

95% CI obtained via the GBNIPC method is compared to experimental results at various stations. Note the 95% confidence intervals were constructed via the bootstrap method using the PC coefficients. The uncertainty band of the pressure coefficient is too narrow to account for the observed discrepancy between prediction and experiment. Moreover, the level of uncertainty in the angle-of-attack and Mach-number do not account for the observed discrepancies between the predictions and experiments. This observation indicates that the observed discrepancies are likely due to the CFD model uncertainty.

6.3 Turbulence Model Uncertainty Analysis

In addition to the uncertainties associated with CFD model inputs, there are often uncertainties associated with turbulence modeling. The uncertainty of turbulence modeling is manifested in the large variety of available models in the literature. Hence, it is important to address turbulence model uncertainty. This information not only gives an assessment to the importance and contribution of each source of uncertainty, but also an indication to where the available computational resources should be focused.

The one-equation S-A turbulence model[74], which assumes the Boussinesq hypothesis, is address in Chapter 5. It is a relatively simple model that has been optimized for aerodynamic applications, most notably for flow past a wing. This model is based on a postulated transport equation comprised of terms modeling convection, diffusion, production, and dissipation of turbulence eddy viscosity. One of the main advantage of the S-A model is the simplicity in imposing the free-stream and wall boundary conditions.

Turbulence model uncertainty analysis is performed on a compressible, transonic,

viscous, fully turbulent flow around the Onera-M6-Wing. This analysis is accomplished by performing Complex Variable sensitivity analysis coupled with the Collocation method on the incoming turbulence and closure coefficients of the S-A turbulence model.

Sensitivity results indicate that the lift and drag coefficients are not highly sensitive to the macrostructure scale of the turbulent flow(i.e., the large, energy rich eddies). As expected, the lift and drag coefficients are also not highly sensitive to the incoming turbulence at the inlet, since the turbulence model is optimized for aerodynamic applications. However, the lift and drag coefficients are highly sensitive to the microstructure scale(i.e., the wall function and the production of turbulence). Lastly, the lift coefficient is highly sensitive to the coefficient of the eddy viscosity C_{v1} in contrast to the drag coefficient. The above observations suggest focusing resources in the refinement of the S-A turbulence model in the microstructure scale.

Although the total number of parameters in the S-A turbulence model is eight for this application, uncertainty is only introduced to the parameters with the strongest influence on the lift and drag coefficient, namely, C_{v1} , C_{b1} , σ , and κ . These parameters, which correspond to the coefficient and the production of turbulence viscosity and wall function, depend among other factors for which no information is available. The turbulence model uncertainty caused by missing variables is introduced by assigning uniform distributions to these parameters. These are reasonable distributions to choose when other distributions are unknown for the problem at hand. In addition, the entire uncertainty interval is contained in a uniform random variable(e.g., the maximum bound).

The mean lift and drag coefficients in concert with their 95% confidence intervals obtained via the Collocation method is compared to experimental results. Note the 95% confidence intervals were constructed via the bootstrap method using the PC coefficients. A 3rd order chaos with four random dimensions required 35 deterministic

runs for both cases. The uncertainty band of the lift of coefficient is too narrow to account for the observed discrepancy between prediction and experiment. Furthermore, the level of uncertainty in the closure coefficients seems to have some effect on the drag coefficient. However, the uncertainty band in both the lift and drag coefficients does not overlap the experimental uncertainty band.

These observations are believed to be due to the differences in computing the uncertainty band between the experiment and prediction. For example, uncertainty band of the experimental measurement is an estimate of the difference between the true and predicted solution in contrast to the CI. In addition, the uncertainty of the closure coefficients does not have a large effect on the lift and drag coefficient since the S-A turbulence model is optimized for this aerodynamic application.

6.4 Future Work

The NIPC methods developed and implemented in this thesis addressed only random variables, i.e., random quantities that do not vary with time or space. Random quantities that vary with time or space can be regarded as an infinite(or finite) set of random variables or random processes. From the perspective of the uncertainties that occur in CFD models, the random variables in CFD parameters may vary with time or space. For problems involving random processes(*i.e.*, many random variables), NIPC methods generally required a large number of deterministic runs which may be prohibitively expensive. For this reason, research should refocus on the NIPC method for problems involving random processes. This may include the investigation of adaptive sampling techniques for efficient calculation of PC coefficients that are important to the response of interest.

Lastly, since the level of uncertainty in numerical simulations is dependent upon grid quality, it is essential to perform comprehensive grid convergence studies. More

specifically, further research should focus on the effect of nonuniform grid refinement to the convergence rate for unstructured meshes.

Bibliography

- [1] Daniel P. Aeschliman and William L. Oberkampf. Experimental methodology for computational fluid dynamics code validation. *AIAA Journal*, 36(5):733–741, May 1998.
- [2] AIAA Applied Aerodynamics Technical Committee. *Drag Prediction Workshop, 19th APA Conference*, Anaheim, CA, 2001. <http://www.aiaa.org/tc/apa/dragpredworkshop/dpw.html>.
- [3] Guide for the verification and validation of computational fluid dynamics simulations. AIAA Guide G-077-1998, 1998.
- [4] G. Alefeld and J. Herzberger. *Introduction to Interval Computations*. Academic Press, New York, 1983.
- [5] D. A. Anderson, J. C. Tannehill, and R. H. Pletcher. *Computational Fluid Mechanics and Heat Transfer*. Series in computational methods in mechanics and thermal sciences. McGraw-Hill, 1997.
- [6] John D. Anderson. *Modern Compressible Flow*. McGraw-Hill, New York, USA, 1982.
- [7] P. Pebay O. Knio R. Ghanem B. Debusschere, H. Najm and O. Le Maitre. Numerical challenges in the use of polynomial chaos representation for stochastic processes. *SIAM Journal of Scientific Computing*, 26, 2004.
- [8] B. Baldwin and H. Lomax. Thin layer approximation and algebraic model for separated turbulent flow, AIAA-paper 1978-0257. 1978.

- [9] B.S. Baldwin and T.J. Barth. A one-equation turbulence transport model for high reynolds number wall-bounded flows, AIAA-paper 1991-0610. Reno, NV, 1991.
- [10] T. J. Barber. Role of code validation and certification in the design environment. *AIAA Journal*, 36(5):752–758, May 1998.
- [11] John A. Benek, Edward M. Kraft, and Rodney F. Lauer. Validation issues for engine-airframe integration. *AIAA Journal*, 36(5):759–764, May 1998.
- [12] C. H. Bischof, G. F. Corliss, L. L. Green, A. Griewank, K. Haigler, and P. A. Newman. Automatic differentiation of advanced CFD codes for multidisciplinary design. *Journal on Computing Systems in Engineering*, 3(6):625–637, May 1993.
- [13] C.H. Bischof, A. Carle, P. Khademi, and A. Mauer. *ADIFOR 2.0 - automatic differentiation of Fortran 77 programs*. IEEE Computational Science & Engineering., 3(3):18-32, 1996.
- [14] J.T. Borggaard. *The Sensitivity Equation Method for Optimal Design*. PhD thesis, Virginia Tech, 1994.
- [15] F.H. Clauser. *Turbulent Boundary Layer*. in Advances in Applied Mechanics. Academic Press, New York, 1956.
- [16] J. Dacles-Mariani, G.G. Zilliac, J.S. Chow, and P. Bradshaw. Numerical/experimental study of a wing tip vortex in the near field. *AIAA Journal*, 33:1561–1568, 1995.
- [17] Germund Dahlquist and Ake Bjorck. *Numerical Methods*. Series in computational methods in mechanics and thermal sciences. Prentice-Hall, 1974.
- [18] L. Devroye. *Non-Uniform Random Variate Generation*. John Wiley & Sons, Springer-Verlag, New York, 1986.
- [19] J.D. Doll and D.L. Freeman. Randomly exact methods. *Science*, 234:1356–1360, 1986.
- [20] David S. Dolling. High-speed turbulent separated flows: Consistency of mathematical models and flow physics. *AIAA Journal*, 36(5):725–732, May 1998.

- [21] E.R. Van Driest. On turbulent flow near a wall. *Aerospace Science*, 23:1007–1012, 1956.
- [22] D.Pelletier E. Turgeon and J. Borggaard. Sensitivity and uncertainty analysis for variable property flows, AIAA-paper 2001-0140. In *39th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January, 2001. CD-ROM.
- [23] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, New York, NY, USA, 1993.
- [24] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, New York, 1993.
- [25] Victor E. Garzon and David L. Darmofal. Using computational fluid dynamics in probabilistic engineering design, AIAA paper 2001-2526. In *15th AIAA Computational Fluid Dynamics Conference*, June 11-14, 2001.
- [26] Roger Ghanem. Stochastic finite elements with multiple random non-gaussian properties. *Journal of Engineering Mechanics*, pages 26–40, January 1999.
- [27] Roger Ghanem and P. D. Spanos. Polynomial chaos in stochastic finite elements. *Journal of Applied Mechanics*, 57:197–202, March 1990.
- [28] Roger G. Ghanem. Ingredients for a general purpose stochastic finite element formulation. *Computational Methods in Applied Mechanical Engineering*, 168:19–34, 1999.
- [29] Roger G. Ghanem and P. D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Springer-Verlag, New York, 1991.
- [30] Andrew G. Godfrey. *Topics on Spatially High-Order Accurate Methods and Preconditioning for the Navier-Stokes Equations with Finite-Rate Chemistry*. PhD thesis, Virginia Tech, 1992.
- [31] Andrew G. Godfrey and Eugene M. Cliff. Sensitivity equations for turbulent flows, AIAA-paper 2001-1060. In *39th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January, 2001. CD-ROM.

- [32] L. L. Green, P. A. Newman, and K. Haigler. Sensitivity derivatives for advanced CFD algorithm and viscous modeling parameters via automatic differentiation. *Journal of Computational Physics*, 125:313–324, 1996.
- [33] A. Griewank and Eds. Corliss, G. *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*, SIAM, 1991.
- [34] W. G. Habashi, J. Dompierre, Y. Bourgault, M. Fortin, and M.G. Wallet. Certifiable computational fluid dynamics through mesh optimization. *AIAA Journal*, 36(5):703–711, May 1998.
- [35] Arthur W. Hafner. *Descriptive Statistical Techniques*. American Library Association, 1989.
- [36] Achintya Haldar and Sankaran Mahadevan. *Probability, Reliability and Statistical Methods in Engineering Design*. John Wiley and Sons, Inc., 2000.
- [37] A. Hall. On an experimental determination of π . *Messeng. Math.*, 2:113–114, 1873.
- [38] J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods*. Methuen’s Monographs on Applied Probability and Statistics. Fletcher & Son Ltd., Norwich, 1964.
- [39] J. C. Helton and F. J. Davis. Sampling-based methods for uncertainty and sensitivity analysis. Technical Report SAND99-2240, Sandia National Laboratories, July 2000.
- [40] J. O. Hinze. *Turbulence, 2nd Edition*. New York, NY, 1975.
- [41] Grossman B. Watson L.T. Mason W.H. Hosder, S. and R.T. Hafka. Observations on cfd simulation uncertainties, AIAA-paper 2002-5531. In 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimazation, Atlanta, GA, September, 2002. CD-ROM.
- [42] Walters R.W Hosder, S. and R. Perez. A non-intrusive implicit polynomial chaos method for uncertianty propagation in cfd simulations, AIAA-paper 2006-0891. In 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January, 2006. CD-ROM.

- [43] L. Huyse. Free-form airfoil shape optimization under uncertainty using maximum expected value and second-order second-moment strategies. Technical report, ICASE Report 2001-18/NASA CR 2001-211020, 2001.
- [44] L. Huyse. Solving Problems of Optimization Under Uncertainty as Statistical Decision Problems, AIAA-Paper 2001-1519. In *42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit*, Seattle, WA, 2001.
- [45] L. Huyse and R. Michael Lewis. Aerodynamic shape optimization of two-dimensional airfoils under uncertain conditions. Technical report, ICASE Report 2001-1/NASA CR 2001-210648, 2001.
- [46] L. Huyse and R. W. Walters. Random field solutions including boundary condition uncertainty for the steady-state generalized burger equation. Technical report, ICASE Report 2001-35/NASA CR 2001-211239, 2001.
- [47] Sastry S. Isukapalli. *Uncertainty Analysis of Transport-Transformation Models*. PhD thesis, The State University of New Jersey, 1999.
- [48] D. Pelletier S. Etienne J. Borggaard, E. Turgeon. Sensitivity and uncertainty analysis for turbulent flows, AIAA-paper 2002-0985. In *40th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January, 2002. CD-ROM.
- [49] Anthony Jameson and Luigi Martinelli. Mesh refinement and modeling errors in flow simulations. *AIAA Journal*, 35(5):676–686, May 1998.
- [50] M.H. Kalos and P.A. Whitlock. *Monte Carlo Methods: Basics*. John Wiley & Sons, New York, 1986.
- [51] Lord Kelvin. Nineteenth century clouds over the dynamical theory of heat and light. *Phil. Mag.*, 2(6):1–40, 1901.
- [52] T.A. Zang L. Mathelin, M.Y. Hussaini and F. Bataille. Uncertainty propagation for turbulent, compressible flow in a quasi-1d nozzle using stochastic methods. to appear 2004.

- [53] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods of selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245, 1979.
- [54] Michael D. Mckay. Latin hypercube sampling as a tool in uncertainty analysis of computer models. In *the Proceedings of the Winter Simulation Conference*, December 1992.
- [55] S. McMillin, J. Pittman, and J. Thomas. Computational study of incipient leading-edge separation on a supersonic delta wing. *Journal of Aircraft*, 29(2):203–209, 1992.
- [56] Unmeel B. Mehta. Credible computational fluid dynamics simulations. *AIAA Journal*, 36(5):665–667, May 1998.
- [57] Eric J. Nielsen. *Aerodynamic Design Sensitivities on an Unstructured Mesh Using the Navier-Stokes Equations and a Discrete Adjoint Formulation*. PhD thesis, Virginia Tech, December 1998.
- [58] Eric J. Nielsen and William L. Kleb. Efficient construction of discrete adjoint operators on unstructured grids by using complex variables, AIAA-paper 2005-0324. In *43rd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January, 2005. CD-ROM.
- [59] William L. Oberkampf and Frederick G. Blottner. Issues in computational fluid dynamics code verification and validation. *AIAA Journal*, 36(5):687–695, May 1998.
- [60] W.L. Oberkampf, J.C. Helton, and K. Sentz. Mathematical Representation of Uncertainty, AIAA Paper 2001-1645. In *42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit*, Seattle, WA, 2001. CD-ROM.
- [61] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGrawHill, New York, 1991.

- [62] R. Perez and R. Walters. An implicit compact polynomial chaos formulation for the euler equations, AIAA-paper 2005-1406. In *43th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January, 2005. CD-ROM.
- [63] M. M. Putko, P. A. Newman, A. C. Taylor III, and L. L. Green. Approach for Uncertainty Propagation and Robust Design in CFD Using Sensitivity Derivatives, AIAA Paper 2001-2528. In *AIAA 15th Computational Fluid Dynamics Conference*, Anaheim, CA, June 2001.
- [64] Helen L. Reed, Tim S. Haynes, and William S. Saric. Computational fluid dynamics validation issues in transition modeling. *AIAA Journal*, 36(5):742–751, May 1998.
- [65] Arthur Rizzi and Jan Vos. Toward establishing credibility in computational fluid dynamic simulations. *AIAA Journal*, 36(5):668–675, May 1998.
- [66] Patrick J. Roache. Quantification of uncertainty in computational fluid dynamics. *Annual Review of Fluid Mechanics*, 29:123–160, 1997.
- [67] Patrick J. Roache. Verification of codes and calculations. *AIAA Journal*, 36(5):696–702, May 1998.
- [68] David G. Robinson. Survey of probabilistic methods used in reliability, risk, and uncertainty analysis: Analytical techniques. Technical Report SAND-98-1189, Sandia National Laboratories, June 1998.
- [69] Christopher. J. Roy. Grid Convergence Error Analysis for Mixed-Order Numerical Schemes, AIAA Paper 2001-2606. In *AIAA 15th Computational Fluid Dynamics Conference*, Anaheim, CA, June 2001.
- [70] R. Y. Rubinstein. *Simulation and the Monte Carlo Method*. John Wiley & Sons, New York, 1981.
- [71] Manuel D. Salas. Some observation on grid convergence. *Computer & fluids*, Available online at www.sciencedirect.com, 35:688–692, March 2006.
- [72] V. Schmitt and F. Charpin. Pressure distributions on the onera-m6-wing at transonic mach number. Technical report, AGARD Advisory Report 138, May 1979.

- [73] A. Smith and T. Cebeci. Numerical solution of the turbulent boundary-layer equations. Technical Report 33735, Douglas Aircraft Division Report DAC, 1967.
- [74] P.R. Spalart and S.R. Allmaras. A one-equation turbulence model for aerodynamic flows, AIAA-paper 1992-0439. Reno, NV, 1992.
- [75] Dawn L. Stewart. *Numerical Methods for Accurate Computational of Design Sensitivities*. PhD thesis, Virginia Tech, 1998.
- [76] A. C. Taylor, L. L. Green, P. A. Newman, and M. M. Putko. Some Advanced Concepts in Discrete Aerodynamic Sensitivity Analysis, AIAA Paper 2001-2529. In *AIAA 15th Computational Fluid Dynamics Conference*, Anaheim, CA, June 2001.
- [77] J. Thompson, Z.U.A Warsi, and C.W. Mastin. *Numerical Grid Generation: Foundations and Applications*. North-Holland, 1985.
- [78] C.P. Tsokos. *Probability Distribution: An Introduction to Probability Theory with Applications*. Wadsworth Publishing Company, Belmont, CA, 1972.
- [79] J. Villadsen and M.L. Michelsen. *Solution of Differential Equation Models by Polynomial Approximation*. Prentice-Hall, Englewood Cliffs, New Jersey, 1978.
- [80] J. Helton W. Oberkampf and K. Sentz. Mathematical representation of uncertainty, AIAA-paper 2001-1645. In *42th AIAA/ASME/ASCE/AHS/ASC Structures Dynamics, and Materials Conference and Exhibit*, Seattle, WA, January 2001. CD-ROM.
- [81] R. W. Walters and L. Huyse. Stochastic methods for fluid mechanics - an introduction. Technical Report in preparation, ICASE, NASA Langley Research Center, Hampton, VA, 2001.
- [82] R.W. Walters. Towards stochastic fluid mechanics via polynomial chaos-invited. *AIAA, paper 2003-0413*, 2003.
- [83] R.W. Walters. Optimization and reconstruction via the polynomial chaos. In *Proceedings of the Sixth European Conference on Structural Dynamics*, Paris, France, September, 2005. CD-ROM.

- [84] R.W. Walters and R. Perez. Comments on the formulation of polynomial chaos problems. *9th ASCE EMD/SEI/GI/AD Joint Specialty Conference on Probabilistic Mechanics and Structural Reliability (PMC2004)*, July 2004.
- [85] R.W. Walters and R. Perez. Polynomial chaos in fluid mechanics phase 2 final report. *NASA Grant NAG-1-03028*, March 2004.
- [86] F.M. White. *Viscous Fluid Flow*. ISBN 0-07-069710-8. McGraw-Hill, Inc, 1974.
- [87] Frank M. White. *Viscous Fluid Flow*. McGraw-Hill, Inc., 1974.
- [88] N. Wiener. The homogenous chaos. *American Journal of Mathematics*, 60:897–936, 1938.
- [89] N. Wiener and A. Wintner. The discrete chaos. *American Journal of Mathematics*, 65:279–298, 1943.
- [90] David C. Wilcox. *Turbulence Modeling for CFD*. DCW Industries, Inc., La Canada, CA, second edition, 1998.
- [91] David C. Wilcox. Turbulence modeling: An overview, AIAA-paper 2001-0724. In *39th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January, 2001. CD-ROM.
- [92] S. Wolfram. *The Mathematica Book*. Wolfram Media, Cambridge University Press, fourth edition, 1999.
- [93] D. Xiu and G. E. Karniadakis. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, 2001.
- [94] D. Xiu and G. E. Karniadakis. The *Wiener – Askey* polynomial chaos for stochastic differential equations. Technical Report 01-1, Center for Fluid Mechanics, Division of Applied Mathematics, Brown University, Providence, RI, 2001.
- [95] H. C. Yee and P. K. Sweby. Aspects of numerical uncertainties in time marching to steady-state numerical solutions. *AIAA Journal*, 36(5):712–724, May 1998.

Appendix A

A.1 The Definition of the Operator Form, \oplus

Consider two block matrices defined as,

$$\mathbf{A}_{n,m,l} \equiv \begin{bmatrix} \{\vec{a}_{111}, \vec{a}_{112}, \dots, \vec{a}_{11l}\} & \{\vec{a}_{121}, \vec{a}_{122}, \dots, \vec{a}_{12l}\} & \cdots & \{\vec{a}_{1m1}, \vec{a}_{1m2}, \dots, \vec{a}_{1ml}\} \\ \{\vec{a}_{211}, \vec{a}_{212}, \dots, \vec{a}_{21l}\} & \{\vec{a}_{221}, \vec{a}_{222}, \dots, \vec{a}_{22l}\} & \cdots & \{\vec{a}_{2m1}, \vec{a}_{2m2}, \dots, \vec{a}_{2ml}\} \\ \vdots & \vdots & \ddots & \vdots \\ \{\vec{a}_{n11}, \vec{a}_{n12}, \dots, \vec{a}_{n1l}\} & \{\vec{a}_{n21}, \vec{a}_{n22}, \dots, \vec{a}_{n2l}\} & \cdots & \{\vec{a}_{nm1}, \vec{a}_{nm2}, \dots, \vec{a}_{nml}\} \end{bmatrix}$$

$$\mathbf{B}_{n,m,l} \equiv \begin{bmatrix} \{\vec{b}_{111}, \vec{b}_{112}, \dots, \vec{b}_{11l}\} & \{\vec{b}_{121}, \vec{b}_{122}, \dots, \vec{b}_{12l}\} & \cdots & \{\vec{b}_{1m1}, \vec{b}_{1m2}, \dots, \vec{b}_{1ml}\} \\ \{\vec{b}_{211}, \vec{b}_{212}, \dots, \vec{b}_{21l}\} & \{\vec{b}_{221}, \vec{b}_{222}, \dots, \vec{b}_{22l}\} & \cdots & \{\vec{b}_{2m1}, \vec{b}_{2m2}, \dots, \vec{b}_{2ml}\} \\ \vdots & \vdots & \ddots & \vdots \\ \{\vec{b}_{n11}, \vec{b}_{n12}, \dots, \vec{b}_{n1l}\} & \{\vec{b}_{n21}, \vec{b}_{n22}, \dots, \vec{b}_{n2l}\} & \cdots & \{\vec{b}_{nm1}, \vec{b}_{nm2}, \dots, \vec{b}_{nml}\} \end{bmatrix}$$

where the elements of each block matrix above, denoted by the curly brackets, are a list of vector arrays or scalars or a combination of both (*i.e.*, \vec{a}_{ijk} can be a vector array or a scalar or a combination of both, where $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$, $k = 1, 2, \dots, l$). Note that n, m , and l are integer numbers. Hence, the operator, \oplus , is defined as,

$$\mathbf{A}_{n,m,l} \oplus \mathbf{B}_{n,m,l} \equiv \begin{bmatrix} c_{11l} & c_{12l} & \cdots & c_{1ml} \\ c_{21l} & c_{22l} & \cdots & c_{2ml} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1l} & c_{n2l} & \cdots & c_{nml} \end{bmatrix}$$

where,

$$\begin{aligned}
c_{11l} &= \vec{a}_{111} \cdot \vec{b}_{111} + \vec{a}_{112} \cdot \vec{b}_{112} + \cdots + \vec{a}_{11l} \cdot \vec{b}_{11l} \\
c_{12l} &= \vec{a}_{121} \cdot \vec{b}_{121} + \vec{a}_{122} \cdot \vec{b}_{122} + \cdots + \vec{a}_{12l} \cdot \vec{b}_{12l} \\
&\vdots \\
c_{1ml} &= \vec{a}_{1m1} \cdot \vec{b}_{1m1} + \vec{a}_{1m2} \cdot \vec{b}_{1m2} + \cdots + \vec{a}_{1ml} \cdot \vec{b}_{1ml} \\
c_{21l} &= \vec{a}_{211} \cdot \vec{b}_{211} + \vec{a}_{212} \cdot \vec{b}_{212} + \cdots + \vec{a}_{21l} \cdot \vec{b}_{21l} \\
c_{22l} &= \vec{a}_{221} \cdot \vec{b}_{221} + \vec{a}_{222} \cdot \vec{b}_{222} + \cdots + \vec{a}_{22l} \cdot \vec{b}_{22l} \\
&\vdots \\
c_{2ml} &= \vec{a}_{2m1} \cdot \vec{b}_{2m1} + \vec{a}_{2m2} \cdot \vec{b}_{2m2} + \cdots + \vec{a}_{2ml} \cdot \vec{b}_{2ml} \\
c_{n1l} &= \vec{a}_{n11} \cdot \vec{b}_{n11} + \vec{a}_{n12} \cdot \vec{b}_{n12} + \cdots + \vec{a}_{n1l} \cdot \vec{b}_{n1l} \\
c_{n2l} &= \vec{a}_{n21} \cdot \vec{b}_{n21} + \vec{a}_{n22} \cdot \vec{b}_{n22} + \cdots + \vec{a}_{n2l} \cdot \vec{b}_{n2l} \\
&\vdots \\
c_{nml} &= \vec{a}_{nm1} \cdot \vec{b}_{nm1} + \vec{a}_{nm2} \cdot \vec{b}_{nm2} + \cdots + \vec{a}_{nml} \cdot \vec{b}_{nml}
\end{aligned}$$

For demonstration purposes, let $n = 4$, $m = 1$, $l = 2$, and write the two matrices as,

$$\mathbf{A}_{4,1,2} = \begin{bmatrix} \{\vec{a}_{111}, \vec{a}_{112}\} \\ \{\vec{a}_{211}, \vec{a}_{212}\} \\ \{\vec{a}_{311}, \vec{a}_{312}\} \\ \{\vec{a}_{411}, \vec{a}_{412}\} \end{bmatrix}$$

$$\mathbf{B}_{4,1,2} = \begin{bmatrix} \{\vec{b}_{111}, \vec{b}_{112}\} \\ \{\vec{b}_{211}, \vec{b}_{212}\} \\ \{\vec{b}_{311}, \vec{b}_{312}\} \\ \{\vec{b}_{411}, \vec{b}_{412}\} \end{bmatrix}$$

and use the definition of the operator, \oplus , to write $\mathbf{A}_{4,1,2} \oplus \mathbf{B}_{4,1,2}$ as,

$$\begin{bmatrix} \{\vec{a}_{111}, \vec{a}_{112}\} \\ \{\vec{a}_{211}, \vec{a}_{212}\} \\ \{\vec{a}_{311}, \vec{a}_{312}\} \\ \{\vec{a}_{411}, \vec{a}_{412}\} \end{bmatrix} \oplus \begin{bmatrix} \{\vec{b}_{111}, \vec{b}_{112}\} \\ \{\vec{b}_{211}, \vec{b}_{212}\} \\ \{\vec{b}_{311}, \vec{b}_{312}\} \\ \{\vec{b}_{411}, \vec{b}_{412}\} \end{bmatrix} = \begin{bmatrix} \vec{a}_{111} \cdot \vec{b}_{111} + \vec{a}_{112} \cdot \vec{b}_{112} \\ \vec{a}_{211} \cdot \vec{b}_{211} + \vec{a}_{212} \cdot \vec{b}_{212} \\ \vec{a}_{311} \cdot \vec{b}_{311} + \vec{a}_{312} \cdot \vec{b}_{312} \\ \vec{a}_{411} \cdot \vec{b}_{411} + \vec{a}_{412} \cdot \vec{b}_{412} \end{bmatrix}$$

Appendix B

B.1 Full Flux Jacobian Matrix in Deterministic Form

The deterministic Jacobian matrix of the full flux vector in local coordinates is given by,

$$\frac{\partial \mathcal{F}}{\partial q} = \begin{pmatrix} \bar{u} & n_x \rho & n_y \rho & 0 \\ \bar{u}u & \rho(\bar{u} + un_x) & \rho un_y & n_x \\ \bar{u}v & \rho vn_x & \rho(\bar{u} + vn_y) & n_y \\ \bar{u}q^2/2 & \rho(n_x h_0 + \bar{u}u) & \rho(n_y h_0 + \bar{u}v) & \frac{\gamma \bar{u}}{\gamma - 1} \end{pmatrix} \Delta s \quad (\text{B.1})$$

B.2 Full Flux Jacobian Matrix in Standard PC Form

Deriving the full flux Jacobian Matrix in the Standard PC form is straightforward. We start with Eqn. (3.21) and form the Jacobian by taking the appropriate derivatives,

which yields

$$\begin{aligned}
\frac{\partial f_{1,r}}{\partial \rho_i} &= \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} \delta_{lk} \bar{u}_{ij} \langle \Psi_i \Psi_j \Psi_k \Psi_r \rangle \\
\frac{\partial f_{1,r}}{\partial u_i} &= \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} (n_x \Delta s)_i \delta_{lj} \langle \Psi_i \Psi_j \Psi_k \Psi_r \rangle \\
\frac{\partial f_{1,r}}{\partial v_i} &= \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} (n_y \Delta s)_i \delta_{lj} \langle \Psi_i \Psi_j \Psi_k \Psi_r \rangle \\
\frac{\partial f_{1,r}}{\partial p_l} &= 0 \\
\frac{\partial f_{2,r}}{\partial \rho_i} &= \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} \sum_{m=0}^{N_q} \delta_{kl} \bar{u}_{ij} u_m \langle \Psi_i \Psi_j \Psi_k \Psi_m \Psi_r \rangle \\
\frac{\partial f_{2,r}}{\partial u_i} &= \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} \sum_{m=0}^{N_q} \rho_k [\bar{u}_{ij} \delta_{lm} + (n_x \Delta s)_i u_m \delta_{jl}] \langle \Psi_i \Psi_j \Psi_k \Psi_m \Psi_r \rangle \\
\frac{\partial f_{2,r}}{\partial v_i} &= \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} \sum_{m=0}^{N_q} \rho_k (n_y \Delta s)_i u_m \delta_{jl} \langle \Psi_i \Psi_j \Psi_k \Psi_m \Psi_r \rangle \\
\frac{\partial f_{2,r}}{\partial p_l} &= \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} (n_x \Delta s)_i \delta_{jl} \langle \Psi_i \Psi_j \Psi_r \rangle \\
\frac{\partial f_{3,r}}{\partial \rho_i} &= \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} \sum_{m=0}^{N_q} \delta_{kl} \bar{u}_{ij} v_m \langle \Psi_i \Psi_j \Psi_k \Psi_m \Psi_r \rangle \\
\frac{\partial f_{3,r}}{\partial u_i} &= \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} \sum_{m=0}^{N_q} \rho_k (n_x \Delta s)_i \delta_{jl} v_m \langle \Psi_i \Psi_j \Psi_k \Psi_m \Psi_r \rangle \\
\frac{\partial f_{3,r}}{\partial v_i} &= \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} \sum_{m=0}^{N_q} \rho_k [\bar{u}_{ij} \delta_{lm} + (n_y \Delta s)_i v_m \delta_{jl}] \langle \Psi_i \Psi_j \Psi_k \Psi_m \Psi_r \rangle \\
\frac{\partial f_{3,r}}{\partial p_l} &= \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} (n_y \Delta s)_i \delta_{jl} \langle \Psi_i \Psi_j \Psi_r \rangle \\
\frac{\partial f_{4,r}}{\partial \rho_i} &= \frac{1}{2} \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} \sum_{n=0}^{N_q} \sum_{m=0}^{N_q} \delta_{kl} \bar{u}_{ij} q_{nm}^2 \langle \Psi_i \Psi_j \Psi_k \Psi_n \Psi_m \Psi_r \rangle
\end{aligned} \tag{B.2}$$

$$\begin{aligned}
\frac{\partial f_{4,r}}{\partial u_l} &= \frac{\gamma}{\gamma-1} \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} (n_x \Delta s)_i \delta_{jl} p_k \langle \Psi_i \Psi_j \Psi_k \Psi_r \rangle \\
&\quad + \frac{1}{2} \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} \sum_{n=0}^{N_q} \sum_{m=0}^{N_q} \rho_k [(n_x \Delta s)_i \{q_{nm}^2 \delta_{jl} + u_n u_j \delta_{lm} + u_j u_m \delta_{ln}\}] \langle \Psi_i \Psi_j \Psi_k \Psi_l \Psi_m \Psi_r \rangle \\
&\quad + \frac{1}{2} \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} \sum_{n=0}^{N_q} \sum_{m=0}^{N_q} (n_y \Delta)_i v_j (u_n \delta_{mn} + u_m \delta_{ln}) \langle \Psi_i \Psi_j \Psi_k \Psi_l \Psi_m \Psi_r \rangle \\
\frac{\partial f_{4,r}}{\partial v_l} &= \frac{\gamma}{\gamma-1} \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} (n_y \Delta s)_i \delta_{jl} p_k \langle \Psi_i \Psi_j \Psi_k \Psi_r \rangle \\
&\quad + \frac{1}{2} \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} \sum_{n=0}^{N_q} \sum_{m=0}^{N_q} \rho_k [(n_y \Delta s)_i \{q_{nm}^2 \delta_{jl} + v_n v_j \delta_{lm} + v_j v_m \delta_{ln}\}] \langle \Psi_i \Psi_j \Psi_k \Psi_l \Psi_m \Psi_r \rangle \\
&\quad + \frac{1}{2} \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} \sum_{n=0}^{N_q} \sum_{m=0}^{N_q} (n_x \Delta)_i u_j (v_n \delta_{mn} + u_n \delta_{ln}) \langle \Psi_i \Psi_j \Psi_k \Psi_l \Psi_m \Psi_r \rangle \\
\frac{\partial f_{4,r}}{\partial p_l} &= \frac{\gamma}{\gamma-1} \sum_{i=0}^{N_{g1}} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} \delta_{kl} \bar{u}_{ij} \langle \Psi_i \Psi_j \Psi_k \Psi_r \rangle
\end{aligned} \tag{B.3}$$

B.3 Full Flux Jacobian Matrix in Operator Form

The full flux Jacobian matrix can be written in compact notation and operator form, \oplus , by first replacing the summation notation in Eqn. (B.2) to \otimes notation and using the definition of the operator, \oplus , which yields

$$\begin{aligned}
 & \left[\begin{array}{l} \{\delta_{N_q,l}^s \otimes \bar{w}\} \\ \{\delta_{N_q,l}^s \otimes u, 0, 0\} \\ \{\delta_{N_q,l}^s \otimes v, 0, 0\} \\ \{\frac{1}{2}(\delta_{N_q,l}^s \otimes \bar{w}) \otimes q_m^2, 0, 0\} \end{array} \right] \frac{\gamma-1}{\gamma} \left(\rho \otimes (n_x \Delta s \otimes \delta_{N_q,l}^s) \cdot \frac{1}{2}(\rho \otimes (n_x \Delta s \otimes \delta_{N_q,l}^s) \otimes q_m^2) \cdot \frac{1}{2}(\rho \otimes \bar{w}) \otimes \frac{du}{dv_l} \right. \\
 & \left. \begin{array}{l} \{\rho \otimes (n_x \Delta s \otimes \delta_{N_q,l}^s) \cdot 0, 0, 0\} \\ \{\rho \otimes (n_x \Delta s \otimes \delta_{N_q,l}^s) \otimes u + (\rho \otimes \bar{w}) \otimes \delta_{N_q,l}^s, 0, 0\} \\ \{\rho \otimes (n_x \Delta s \otimes \delta_{N_q,l}^s) \otimes v, 0, 0\} \\ \frac{\gamma-1}{\gamma}(\rho \otimes (n_x \Delta s \otimes \delta_{N_q,l}^s) \otimes q_m^2) \cdot \frac{1}{2}(\rho \otimes \bar{w}) \otimes \frac{du}{dv_l} \end{array} \right) \oplus \\
 & \left[\begin{array}{l} \{\langle \bar{\Psi}_{N_q,Ng1,Ng,r} \rangle, 0, 0\} \\ \{\langle \bar{\Psi}_{N_q,Ng1,Ng,Nq,r} \rangle, 0, 0\} \\ \{\langle \bar{\Psi}_{N_q,Ng1,Ng,Nq,Nq,r} \rangle, 0, 0\} \\ \{\langle \bar{\Psi}_{N_q,Ng1,Ng,Nq,Nq,r} \rangle, 0, 0\} \end{array} \right] \left\{ \langle \bar{\Psi}_{N_q,Ng1,Ng,r} \rangle, 0, 0 \right\} \\
 & \left[\begin{array}{l} \{\langle \bar{\Psi}_{N_q,Ng1,Ng,r} \rangle, 0, 0\} \\ \{\langle \bar{\Psi}_{N_q,Ng1,Ng,Nq,r} \rangle, 0, 0\} \\ \{\langle \bar{\Psi}_{N_q,Ng1,Ng,Nq,Nq,r} \rangle, 0, 0\} \\ \{\langle \bar{\Psi}_{N_q,Ng1,Ng,Nq,Nq,r} \rangle, 0, 0\} \end{array} \right] \left\{ \langle \bar{\Psi}_{N_q,Ng1,Ng,r} \rangle, \langle \bar{\Psi}_{N_q,Ng1,Ng,Nq,r} \rangle, \langle \bar{\Psi}_{N_q,Ng1,Ng,Nq,Nq,r} \rangle, \langle \bar{\Psi}_{N_q,Ng1,Ng,Nq,Nq,r} \rangle \right\} \\
 & \left[\begin{array}{l} \{\rho \otimes (n_y \Delta s \otimes \delta_{N_q,l}^s) \cdot 0, 0, 0\} \\ \{\rho \otimes (n_y \Delta s \otimes \delta_{N_q,l}^s) \otimes u + (\rho \otimes \bar{w}) \otimes \delta_{N_q,l}^s, 0, 0\} \\ \{\rho \otimes (n_y \Delta s \otimes \delta_{N_q,l}^s) \otimes v + (\rho \otimes \bar{w}) \otimes \delta_{N_q,l}^s, 0, 0\} \\ \frac{\gamma-1}{\gamma}(\rho \otimes (n_y \Delta s \otimes \delta_{N_q,l}^s) \otimes q_m^2) \cdot \frac{1}{2}(\rho \otimes \bar{w}) \otimes \frac{du}{dv_l} \end{array} \right] \oplus \\
 & \left[\begin{array}{l} \{0, 0, 0\} \\ \{\langle \bar{\Psi}_{N_q,Ng1,Ng,r} \rangle, 0, 0\} \\ \{\langle \bar{\Psi}_{N_q,Ng1,Ng,Nq,r} \rangle, 0, 0\} \\ \{\langle \bar{\Psi}_{N_q,Ng1,Ng,Nq,Nq,r} \rangle, 0, 0\} \end{array} \right] \oplus \\
 & \left[\begin{array}{l} \{0, 0, 0\} \\ \{\langle \bar{\Psi}_{N_q,Ng1,Ng,r} \rangle, 0, 0\} \\ \{\langle \bar{\Psi}_{N_q,Ng1,Ng,Nq,r} \rangle, 0, 0\} \\ \{\langle \bar{\Psi}_{N_q,Ng1,Ng,Nq,Nq,r} \rangle, 0, 0\} \end{array} \right] \left\{ \langle \bar{\Psi}_{N_q,Ng1,Ng,r} \rangle, \langle \bar{\Psi}_{N_q,Ng1,Ng,Nq,r} \rangle, \langle \bar{\Psi}_{N_q,Ng1,Ng,Nq,Nq,r} \rangle \right\} \quad (\text{B.4})
 \end{aligned}$$

where,

$$\begin{aligned}
 \frac{du}{du_l} &= \vec{\delta}_{N_q,l} \otimes u + u \otimes \vec{\delta}_{N_q,l} \\
 \frac{dv}{dv_l} &= \vec{\delta}_{N_q,l} \otimes v + v \otimes \vec{\delta}_{N_q,l} \\
 q_m^2 &= u \otimes u + v \otimes v \\
 \vec{\delta}_{N_q,l} &= (\delta_{0l}, \delta_{1l}, \dots, \delta_{N_q,l})
 \end{aligned}$$

Where δ is Kronecker delta function. Note the similarity between the deterministic Jacobian full flux matrix (i.e., Eqn. B.1) and the deterministic component of the PC Jacobian full flux matrix (i.e., Eqn. B.4). For example, each element in Eqn. (B.1) is replaced by a list which contains vector arrays and zeros sub-elements. In addition, multiplication in the sub-elements are replaced by the \otimes operation where the scalar inputs become vector arrays.

Appendix C

C.1 Van Leer Split Flux Vector(VLSFV) in Deterministic Form

Characteristic based, upwind schemes give rise to additional numerical flux functions, e.g., Van Leer's Flux Vector Splitting, Roe's Flux Difference Split scheme, and other. The components of the Van Leer split fluxes (times the surface area, Δs) in local coordinates are given by

$$\begin{aligned}
 \mathcal{F}_1^\pm &= \frac{\pm \rho a (M \pm 1)^2}{4} \Delta s \\
 \mathcal{F}_2^\pm &= \mathcal{F}_1^\pm \left[\frac{n_x (-\bar{u} \pm 2a)}{\gamma} + u \right] \\
 \mathcal{F}_3^\pm &= \mathcal{F}_1^\pm \left[\frac{n_y (-\bar{u} \pm 2a)}{\gamma} + v \right] \\
 \mathcal{F}_4^\pm &= \mathcal{F}_1^\pm \left[\frac{-(\gamma - 1)\bar{u}^2 \pm 2(\gamma - 1)\bar{u}a + 2a^2}{\gamma^2 - 1} + \frac{u^2 + v^2}{2} \right]
 \end{aligned} \tag{C.1}$$

where $M = \frac{\bar{u}}{a}$ is the Mach number.

C.2 VLSFV in Standard PC Form

The PC expansion for the components of the Van Leer split fluxes require more effort. First of all, and in contrast to the components of the full flux vector, the quantities n_x , n_y and Δs appear individually (i.e. not just in the products $n_x \Delta s$ and

$n_y \Delta s$). Thus separate expansions are required for these three variables. We represent them in the form of Eqn. 4.5 with $i = 0, 1, \dots, N_{g2}$ input modes (user-defined). In addition, all internal (or intermediate) quantities (e.g. sound speed, Mach number, ...) that are functions of q and/or the input variables are also represented in the form of Eqn. 4.5 with $i = 0, 1, \dots, N_{int}$ internal modes (user-defined). Obtaining the expansion coefficients of any intermediate variable is relatively straightforward and typically involves solving a small linear problem. Going through the algebra of substituting PC expansions for the input, output and internal variables into Van Leer's deterministic flux vector splitting given by Eqn. C.1, one obtains the following

components of the Van Leer PC split flux,

$$\begin{aligned}
\mathcal{F}_{1,r}^\pm &= \pm \frac{1}{4} \sum_{i=0}^{N_q} \sum_{j=0}^{N_{int}} \sum_{k=0}^{N_{int}} \sum_{l=0}^{N_{g2}} \rho_i a_j (M \pm 1)_k^2 \Delta s_l \langle \Psi_i \Psi_j \Psi_k \Psi_l \Psi_r \rangle \\
\mathcal{F}_{2,r}^\pm &= \sum_{i=0}^{N_q} \sum_{j=0}^{N_{g2}} \sum_{k=0}^{N_{g2}} \sum_{l=0}^{N_q} (\mathcal{F}_1^\pm)_i n_{x_j} \left(\frac{-\bar{u}_{kl}}{\gamma} \right) \langle \Psi_i \Psi_j \Psi_k \Psi_l \Psi_r \rangle \\
&\quad + \sum_{i=0}^{N_q} \sum_{j=0}^{N_{g2}} \sum_{k=0}^{N_{int}} (\mathcal{F}_1^\pm)_i n_{x_j} \left(\frac{\pm 2a_k}{\gamma} \right) \langle \Psi_i \Psi_j \Psi_k \Psi_r \rangle \\
&\quad + \sum_{i=0}^{N_q} \sum_{j=0}^{N_q} (\mathcal{F}_1^\pm)_i u_j \langle \Psi_i \Psi_j \Psi_r \rangle \\
\mathcal{F}_{3,r}^\pm &= \sum_{i=0}^{N_q} \sum_{j=0}^{N_{g2}} \sum_{k=0}^{N_{g2}} \sum_{l=0}^{N_q} (\mathcal{F}_1^\pm)_i n_{y_j} \left(\frac{-\bar{u}_{kl}}{\gamma} \right) \langle \Psi_i \Psi_j \Psi_k \Psi_l \Psi_r \rangle \\
&\quad + \sum_{i=0}^{N_q} \sum_{j=0}^{N_{g2}} \sum_{k=0}^{N_{int}} (\mathcal{F}_1^\pm)_i n_{y_j} \left(\frac{\pm 2a_k}{\gamma} \right) \langle \Psi_i \Psi_j \Psi_k \Psi_r \rangle \\
&\quad + \sum_{i=0}^{N_q} \sum_{j=0}^{N_q} (\mathcal{F}_1^\pm)_i v_j \langle \Psi_i \Psi_j \Psi_r \rangle \\
\mathcal{F}_{4,r}^\pm &= \left\{ \sum_{i=0}^{N_q} \sum_{j=0}^{N_{g2}} \sum_{k=0}^{N_q} \sum_{l=0}^{N_{g2}} \sum_{m=0}^{N_q} (\mathcal{F}_1^\pm)_i \left[-\frac{\gamma-1}{\gamma^2-1} \bar{u}_{jk} \bar{u}_{lm} \right] \right\} \langle \Psi_i \Psi_j \Psi_k \Psi_l \Psi_m \Psi_r \rangle \\
&\quad \pm \left\{ \sum_{i=0}^{N_q} \sum_{j=0}^{N_{g2}} \sum_{k=0}^{N_q} \sum_{l=0}^{N_{int}} (\mathcal{F}_1^\pm)_i \left[\frac{2}{\gamma^2-1} \bar{u}_{jk} a_l \right] \right\} \langle \Psi_i \Psi_j \Psi_k \Psi_l \Psi_r \rangle \\
&\quad + \left\{ \sum_{i=0}^{N_q} \sum_{j=0}^{N_{int}} \sum_{k=0}^{N_{int}} (\mathcal{F}_1^\pm)_i \left[\frac{2}{\gamma^2-1} a_{jk}^2 \right] \right\} \langle \Psi_i \Psi_j \Psi_k \Psi_r \rangle \\
&\quad + \left\{ \sum_{i=0}^{N_q} \sum_{j=0}^{N_q} \sum_{k=0}^{N_q} (\mathcal{F}_1^\pm)_i \frac{q_{jk}^2}{2} \right\} \langle \Psi_i \Psi_j \Psi_k \Psi_r \rangle \tag{C.2}
\end{aligned}$$

where

$$\begin{aligned}
q_{jk}^2 &= u_j u_k + v_j v_k \\
\bar{u}_{kl} &= (n_x)_k u_l + (n_y)_k v_l.
\end{aligned}$$

Note that PC representation is not unique. For example, the quantity $(M \pm 1)^2$ in the form of Eqn. 4.5 was expanded, although $M \pm 1$ or M could have been expanded alternatively. Further, note that the second through fourth components of the split flux vector depend on *all modes* of the first component. This has algorithmic consequences in the implementation.

C.3 VLSFV in PC Compact Form

The components of the Van Leer split fluxes given by Eqn. C.2 in short-hand notation are:

$$\begin{aligned}
\mathcal{F}_{1,r}^\pm &= \pm \frac{1}{4} \rho \otimes a \otimes (M \pm 1)^2 \otimes \Delta s \cdot \langle \vec{\Psi}_{N_q, N_{int}, N_{int}, N_{g2}, r} \rangle \\
\mathcal{F}_{2,r}^\pm &= \left[(\mathcal{F}_1^\pm) \otimes n_x \otimes \left(\frac{-\bar{u}}{\gamma} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{g2}, N_q, r} \rangle \\
&\quad + \left[(\mathcal{F}_1^\pm) \otimes n_x \otimes \left(\frac{\pm 2a}{\gamma} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&\quad + [(\mathcal{F}_1^\pm) \otimes u] \cdot \langle \vec{\Psi}_{N_q, N_q, r} \rangle \\
\mathcal{F}_{3,r}^\pm &= \left[(\mathcal{F}_1^\pm) \otimes n_y \otimes \left(\frac{-\bar{u}}{\gamma} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{g2}, N_q, r} \rangle \\
&\quad + \left[(\mathcal{F}_1^\pm) \otimes n_y \otimes \left(\frac{\pm 2a}{\gamma} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&\quad + [(\mathcal{F}_1^\pm) \otimes v] \cdot \langle \vec{\Psi}_{N_q, N_q, r} \rangle \\
\mathcal{F}_{4,r}^\pm &= \left\{ (\mathcal{F}_1^\pm) \otimes \left[-\frac{\gamma-1}{\gamma^2-1} \bar{u} \otimes \bar{u} \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_q, N_{g2}, N_q, r} \rangle \\
&\quad \pm \left\{ (\mathcal{F}_1^\pm) \otimes \left[\frac{2}{\gamma^2-1} \bar{u} \otimes a \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&\quad + \left\{ (\mathcal{F}_1^\pm) \otimes \left[\frac{2}{\gamma^2-1} a \otimes a \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{int}, N_{int}, r} \rangle \\
&\quad + \left\{ (\mathcal{F}_1^\pm) \otimes \left[\frac{u \otimes u + v \otimes v}{2} \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_q, N_q, r} \rangle
\end{aligned} \tag{C.3}$$

where

$$\bar{u} = n_x \otimes u + n_y \otimes v.$$

Note again the similarity between Eqn. C.3 and the deterministic version, Eqn. C.1.

C.4 VLSFV Jacobian Matrix in Deterministic Form

With $V^2 = u^2 + v^2$, the deterministic jacobian Van Leer components of the matrix in local coordiantes is given by;

$$\begin{aligned}
\frac{\partial F_1^\pm}{\partial \rho} &= \pm \frac{a}{8}(M+1)(3M \pm 1) \\
\frac{\partial F_1^\pm}{\partial u} &= \pm \frac{\rho}{2}(M \pm 1)n_x \\
\frac{\partial F_1^\pm}{\partial v} &= \pm \frac{\rho}{2}(M \pm 1)n_y \\
\frac{\partial F_1^\pm}{\partial p} &= \pm \frac{\gamma}{2a}(-M^2 + 1) \\
\frac{\partial F_2^\pm}{\partial \rho} &= F_1^\pm \frac{n_x}{\gamma} \left(\mp \frac{a}{\rho} \right) + \left(\frac{n_x(-\bar{u} \pm 2a)}{\gamma} + u \right) \frac{\partial F_1^\pm}{\partial \rho} \\
\frac{\partial F_2^\pm}{\partial u} &= F_1^\pm \left(1 - \frac{n_x^2}{\gamma} \right) + \left(\frac{n_x(-\bar{u} \pm 2a)}{\gamma} + u \right) \frac{\partial F_1^\pm}{\partial u} \\
\frac{\partial F_2^\pm}{\partial v} &= F_1^\pm \left(-\frac{n_x n_y}{\gamma} \right) + \left(\frac{n_x(-\bar{u} \pm 2a)}{\gamma} + u \right) \frac{\partial F_1^\pm}{\partial v} \\
\frac{\partial F_2^\pm}{\partial p} &= F_1^\pm \left(\pm \frac{n_x a}{\gamma p} \right) + \left(\frac{n_x(-\bar{u} \pm 2a)}{\gamma} + u \right) \frac{\partial F_1^\pm}{\partial p} \\
\frac{\partial F_3^\pm}{\partial \rho} &= F_1^\pm \left(\mp \frac{n_y a}{\gamma \rho} \right) + \left(\frac{n_y(-\bar{u} \pm 2a)}{\gamma} + v \right) \frac{\partial F_1^\pm}{\partial \rho} \\
\frac{\partial F_3^\pm}{\partial u} &= F_1^\pm \left(\frac{-n_x n_y}{\gamma} \right) + \left(\frac{n_y(-\bar{u} \pm 2a)}{\gamma} + v \right) \frac{\partial F_1^\pm}{\partial u} \\
\frac{\partial F_3^\pm}{\partial v} &= F_1^\pm \left(\frac{1 - n_y^2}{\gamma} \right) + \left(\frac{n_y(-\bar{u} \pm 2a)}{\gamma} + v \right) \frac{\partial F_1^\pm}{\partial v} \\
\frac{\partial F_3^\pm}{\partial p} &= F_1^\pm \left(\pm \frac{n_y a}{\gamma p} \right) + \left(\frac{n_y(-\bar{u} \pm 2a)}{\gamma} + v \right) \frac{\partial F_1^\pm}{\partial p} \\
\frac{\partial F_4^\pm}{\partial \rho} &= F_1^\pm \left(-\frac{a}{\rho(\gamma^2 - 1)} [\pm(\gamma - 1)\bar{u} + 2a] \right) + \left(\frac{(1 - \gamma)\bar{u}^2 \pm 2(\gamma - 1)\bar{u}a + 2a^2}{\gamma^2 - 1} + \frac{V^2}{2} \right) \frac{\partial F_1^\pm}{\partial \rho} \\
\frac{\partial F_4^\pm}{\partial u} &= F_1^\pm \left(\frac{2n_x}{\gamma + 1} [-\bar{u} \pm a] + u \right) + \left(\frac{-(\gamma - 1)\bar{u}^2 \pm 2(\gamma - 1)\bar{u}a + 2a^2}{\gamma^2 - 1} + \frac{V^2}{2} \right) \frac{\partial F_1^\pm}{\partial u} \\
\frac{\partial F_4^\pm}{\partial v} &= F_1^\pm \left(\frac{2n_y}{\gamma + 1} [-\bar{u} \pm a] + v \right) + \left(\frac{-(\gamma - 1)\bar{u}^2 \pm 2(\gamma - 1)\bar{u}a + 2a^2}{\gamma^2 - 1} + \frac{V^2}{2} \right) \frac{\partial F_1^\pm}{\partial v} \\
\frac{\partial F_4^\pm}{\partial p} &= F_1^\pm \left(\frac{\gamma}{\rho(\gamma^2 - 1)} [\pm(\gamma - 1)M + 2] \right) + \left(\frac{-(\gamma - 1)\bar{u}^2 \pm 2(\gamma - 1)\bar{u}a + 2a^2}{\gamma^2 - 1} + \frac{V^2}{2} \right) \frac{\partial F_1^\pm}{\partial p}
\end{aligned}$$

C.5 VLSFV Jacobian Matrix in PC Compact Form

$$\begin{aligned}
\frac{\partial \mathcal{F}_{1,r}^\pm}{\partial \rho_l} &= \left[\pm \frac{1}{4} \left(\vec{\delta}_{N_q,l} \otimes a \right) \otimes \left((M \pm 1)^2 \otimes \Delta s \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{int}, N_{int}, N_{g2}, r} \rangle \\
&+ \left[\pm \frac{1}{4} \left(\rho \otimes \frac{\partial a}{\partial \rho_l} \right) \otimes \left((M \pm 1)^2 \otimes \Delta s \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{int}, N_{int}, N_{g2}, r} \rangle \\
&+ \left[\pm \frac{1}{4} \left(\rho \otimes a \right) \otimes \left(\frac{\partial (M \pm 1)^2}{\partial \rho_l} \otimes \Delta s \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{int}, N_{int}, N_{g2}, r} \rangle \\
\frac{\partial \mathcal{F}_{1,r}^\pm}{\partial u_l} &= \left[\pm \frac{1}{4} \left(\rho \otimes a \right) \otimes \left(\frac{\partial (M \pm 1)^2}{\partial u_l} \otimes \Delta s \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{int}, N_{int}, N_{g2}, r} \rangle \\
\frac{\partial \mathcal{F}_{1,r}^\pm}{\partial v_l} &= \left[\pm \frac{1}{4} \left(\rho \otimes a \right) \otimes \left(\frac{\partial (M \pm 1)^2}{\partial v_l} \otimes \Delta s \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{int}, N_{int}, N_{g2}, r} \rangle \\
\frac{\partial \mathcal{F}_{1,r}^\pm}{\partial p_l} &= \left[\pm \frac{1}{4} \left(\rho \otimes \frac{\partial a}{\partial p_l} \right) \otimes \left((M \pm 1)^2 \otimes \Delta s \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{int}, N_{int}, N_{g2}, r} \rangle \\
&+ \left[\pm \frac{1}{4} \rho \otimes a \otimes \frac{\partial (M \pm 1)^2}{\partial p_l} \otimes \Delta s \right] \cdot \langle \vec{\Psi}_{N_q, N_{int}, N_{int}, N_{g2}, r} \rangle \\
\frac{\partial \mathcal{F}_{2,r}^\pm}{\partial \rho_l} &= \left[\left(\frac{\partial (\mathcal{F}_1^\pm)}{\partial \rho_l} \otimes n_x \right) \otimes \left(\frac{-\bar{u}}{\gamma} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{g2}, N_q, r} \rangle \\
&+ \left[\left(\frac{\partial (\mathcal{F}_1^\pm)}{\partial \rho_l} \otimes n_x \right) \otimes \left(\frac{\pm 2a}{\gamma} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&+ \left[\left(\mathcal{F}_1^\pm \otimes n_x \right) \otimes \left(\frac{\pm 2}{\gamma} \frac{\partial a}{\partial \rho} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&+ \left[\frac{\partial (\mathcal{F}_1^\pm)}{\partial \rho_l} \otimes u \right] \cdot \langle \vec{\Psi}_{N_q, N_q, r} \rangle \\
\frac{\partial \mathcal{F}_{2,r}^\pm}{\partial u_l} &= \left[\left(\frac{\partial (\mathcal{F}_1^\pm)}{\partial u_l} \otimes n_x \right) \otimes \left(\frac{-\bar{u}}{\gamma} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{g2}, N_q, r} \rangle \\
&+ \left[\left(\mathcal{F}_1^\pm \otimes n_x \right) \otimes \left(\frac{-1}{\gamma} (n_x \otimes \vec{\delta}_{N_q,l}) \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{g2}, N_q, r} \rangle \\
&+ \left[\left(\frac{\partial (\mathcal{F}_1^\pm)}{\partial u_l} \otimes n_x \right) \otimes \left(\frac{\pm 2a}{\gamma} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&+ \left[\frac{\partial (\mathcal{F}_1^\pm)}{\partial u_l} \otimes u \right] \cdot \langle \vec{\Psi}_{N_q, N_q, r} \rangle \\
&+ \left[\mathcal{F}_1^\pm \otimes \vec{\delta}_{N_q,l} \right] \cdot \langle \vec{\Psi}_{N_q, N_q, r} \rangle \\
\frac{\partial \mathcal{F}_{2,r}^\pm}{\partial v_l} &= \left[\left(\frac{\partial (\mathcal{F}_1^\pm)}{\partial v_l} \otimes n_x \right) \otimes \left(\frac{-\bar{u}}{\gamma} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{g2}, N_q, r} \rangle \\
&+ \left[\left(\mathcal{F}_1^\pm \otimes n_x \right) \otimes \left(\frac{-1}{\gamma} (n_y \otimes \vec{\delta}_{N_q,l}) \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{g2}, N_q, r} \rangle \\
&+ \left[\left(\frac{\partial (\mathcal{F}_1^\pm)}{\partial v_l} \otimes n_x \right) \otimes \left(\frac{\pm 2a}{\gamma} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&+ \left[\frac{\partial (\mathcal{F}_1^\pm)}{\partial v_l} \otimes u \right] \cdot \langle \vec{\Psi}_{N_q, N_q, r} \rangle
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathcal{F}_{2,r}^\pm}{\partial p_l} &= \left[\left(\frac{\partial(\mathcal{F}_1^\pm)}{\partial p_l} \otimes n_x \right) \otimes \left(\frac{-\bar{u}}{\gamma} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{g2}, N_q, r} \rangle \\
&+ \left[\left(\frac{\partial(\mathcal{F}_1^\pm)}{\partial p_l} \otimes n_x \right) \otimes \left(\frac{\pm 2a}{\gamma} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&+ \left[(\mathcal{F}_1^\pm \otimes n_x) \otimes \left(\frac{\pm 2}{\gamma} \frac{\partial a}{\partial p_l} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&+ \left[\frac{\partial(\mathcal{F}_1^\pm)}{\partial p_l} \otimes u \right] \cdot \langle \vec{\Psi}_{N_q, N_q, r} \rangle \\
\frac{\partial \mathcal{F}_{3,r}^\pm}{\partial \rho_l} &= \left[\left(\frac{\partial(\mathcal{F}_1^\pm)}{\partial \rho_l} \otimes n_y \right) \otimes \left(\frac{-\bar{u}}{\gamma} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{g2}, N_q, r} \rangle \\
&+ \left[\left(\frac{\partial(\mathcal{F}_1^\pm)}{\partial \rho_l} \otimes n_y \right) \otimes \left(\frac{\pm 2a}{\gamma} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&+ \left[(\mathcal{F}_1^\pm \otimes n_y) \otimes \left(\frac{\pm 2}{\gamma} \frac{\partial a}{\partial \rho_l} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&+ \left[\frac{\partial(\mathcal{F}_1^\pm)}{\partial \rho_l} \otimes v \right] \cdot \langle \vec{\Psi}_{N_q, N_q, r} \rangle \\
\frac{\partial \mathcal{F}_{3,r}^\pm}{\partial u_l} &= \left[\left(\frac{\partial(\mathcal{F}_1^\pm)}{\partial u_l} \otimes n_y \right) \otimes \left(\frac{-\bar{u}}{\gamma} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{g2}, N_q, r} \rangle \\
&+ \left[(\mathcal{F}_1^\pm \otimes n_y) \otimes \left(\frac{-1}{\gamma} (n_x \otimes \vec{\delta}_{N_q, l}) \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{g2}, N_q, r} \rangle \\
&+ \left[\left(\frac{\partial(\mathcal{F}_1^\pm)}{\partial u_l} \otimes n_y \right) \otimes \left(\frac{\pm 2a}{\gamma} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&+ \left[\frac{\partial(\mathcal{F}_1^\pm)}{\partial u_l} \otimes v \right] \cdot \langle \vec{\Psi}_{N_q, N_q, r} \rangle \\
\frac{\partial \mathcal{F}_{3,r}^\pm}{\partial v_l} &= \left[\left(\frac{\partial(\mathcal{F}_1^\pm)}{\partial v_l} \otimes n_y \right) \otimes \left(\frac{-\bar{u}}{\gamma} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{g2}, N_q, r} \rangle \\
&+ \left[(\mathcal{F}_1^\pm \otimes n_y) \otimes \left(\frac{-1}{\gamma} (n_y \otimes \vec{\delta}_{N_q, l}) \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{g2}, N_q, r} \rangle \\
&+ \left[\left(\frac{\partial(\mathcal{F}_1^\pm)}{\partial v_l} \otimes n_y \right) \otimes \left(\frac{\pm 2a}{\gamma} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&+ \left[\frac{\partial(\mathcal{F}_1^\pm)}{\partial v_l} \otimes v \right] \cdot \langle \vec{\Psi}_{N_q, N_q, r} \rangle \\
&+ \left[\mathcal{F}_1^\pm \otimes \vec{\delta}_{N_q, l} \right] \cdot \langle \vec{\Psi}_{N_q, N_q, r} \rangle \\
\frac{\partial \mathcal{F}_{3,r}^\pm}{\partial p_l} &= \left[\left(\frac{\partial(\mathcal{F}_1^\pm)}{\partial p_l} \otimes n_y \right) \otimes \left(\frac{-\bar{u}}{\gamma} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{g2}, N_q, r} \rangle \\
&+ \left[\left(\frac{\partial(\mathcal{F}_1^\pm)}{\partial p_l} \otimes n_y \right) \otimes \left(\frac{\pm 2a}{\gamma} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&+ \left[(\mathcal{F}_1^\pm \otimes n_y) \otimes \left(\frac{\pm 2}{\gamma} \frac{\partial a}{\partial p_l} \right) \right] \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&+ \left[\frac{\partial(\mathcal{F}_1^\pm)}{\partial p_l} \otimes v \right] \cdot \langle \vec{\Psi}_{N_q, N_q, r} \rangle
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathcal{F}_{4,r}^\pm}{\partial \rho_l} &= \left\{ \frac{\partial(\mathcal{F}_1^\pm)}{\partial \rho_l} \otimes \left[-\frac{\gamma-1}{\gamma^2-1} \bar{u} \otimes \bar{u} \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_q, N_{g2}, N_q, r} \rangle \\
&\pm \left\{ \frac{\partial(\mathcal{F}_1^\pm)}{\partial \rho_l} \otimes \left[\frac{2(\gamma-1)}{\gamma^2-1} \bar{u} \otimes a \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_q, N_{int}, r} \rangle \\
&\pm \left\{ (\mathcal{F}_1^\pm) \otimes \left[\frac{2(\gamma-1)}{\gamma^2-1} \bar{u} \otimes \frac{\partial a}{\partial \rho_l} \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_q, N_{int}, r} \rangle \\
&+ \left\{ \frac{\partial(\mathcal{F}_1^\pm)}{\partial \rho_l} \otimes \left[\frac{2}{\gamma^2-1} a \otimes a \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{int}, N_{int}, r} \rangle \\
&+ \left\{ \mathcal{F}_1^\pm \otimes \left[\frac{2}{\gamma^2-1} \frac{\partial a}{\partial \rho_l} \otimes a \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{int}, N_{int}, r} \rangle \\
&+ \left\{ \mathcal{F}_1^\pm \otimes \left[\frac{2}{\gamma^2-1} a \otimes \frac{\partial a}{\partial \rho_l} \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{int}, N_{int}, r} \rangle \\
&+ \left\{ \frac{(\partial \mathcal{F}_1^\pm)}{\partial \rho_l} \otimes \left[\frac{u \otimes u + v \otimes v}{2} \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_q, N_q, r} \rangle \\
\frac{\partial \mathcal{F}_{4,r}^\pm}{\partial u_l} &= \left\{ \frac{\partial(\mathcal{F}_1^\pm)}{\partial u_l} \otimes \left[-\frac{\gamma-1}{\gamma^2-1} \bar{u} \otimes \bar{u} \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_q, N_{g2}, N_q, r} \rangle \\
&+ \left\{ \mathcal{F}_1^\pm \otimes \left[-\frac{\gamma-1}{\gamma^2-1} (n_x \otimes \vec{\delta}_{N_q, l}) \otimes \bar{u} \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_q, N_{g2}, N_q, r} \rangle \\
&+ \left\{ \mathcal{F}_1^\pm \otimes \left[-\frac{\gamma-1}{\gamma^2-1} \bar{u} \otimes (n_x \otimes \vec{\delta}_{N_q, l}) \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_q, N_{g2}, N_q, r} \rangle \\
&\pm \left\{ \frac{\partial(\mathcal{F}_1^\pm)}{\partial u_l} \otimes \left[\frac{2(\gamma-1)}{\gamma^2-1} \bar{u} \otimes a \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&\pm \left\{ \mathcal{F}_1^\pm \otimes \left[\frac{2(\gamma-1)}{\gamma^2-1} (n_x \otimes \vec{\delta}_{N_q, l}) \otimes a \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&+ \left\{ \frac{\partial(\mathcal{F}_1^\pm)}{\partial u_l} \otimes \left[\frac{2}{\gamma^2-1} a \otimes a \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{int}, N_{int}, r} \rangle \\
&+ \left\{ \frac{(\partial \mathcal{F}_1^\pm)}{\partial u_l} \otimes \left[\frac{u \otimes u + v \otimes v}{2} \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_q, N_q, r} \rangle \\
&+ \left\{ \mathcal{F}_1^\pm \otimes \left[\frac{\vec{\delta}_{N_q, l} \otimes u + u \otimes \vec{\delta}_{N_q, l}}{2} \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_q, N_q, r} \rangle
\end{aligned}$$

(C.4)

$$\begin{aligned}
\frac{\partial \mathcal{F}_{4,r}^\pm}{\partial v_l} &= \left\{ \frac{\partial(\mathcal{F}_1^\pm)}{\partial v_l} \otimes \left[-\frac{\gamma-1}{\gamma^2-1} \bar{u} \otimes \bar{u} \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_q, N_{g2}, N_q, r} \rangle \\
&+ \left\{ \mathcal{F}_1^\pm \otimes \left[-\frac{\gamma-1}{\gamma^2-1} (n_y \otimes \vec{\delta}_{N_q, l}) \otimes \bar{u} \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_q, N_{g2}, N_q, r} \rangle \\
&+ \left\{ \mathcal{F}_1^\pm \otimes \left[-\frac{\gamma-1}{\gamma^2-1} \bar{u} \otimes (n_y \otimes \vec{\delta}_{N_q, l}) \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_q, N_{g2}, N_q, r} \rangle \\
&\pm \left\{ \frac{\partial(\mathcal{F}_1^\pm)}{\partial v_l} \otimes \left[\frac{2(\gamma-1)}{\gamma^2-1} \bar{u} \otimes a \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&\pm \left\{ \mathcal{F}_1^\pm \otimes \left[\frac{2(\gamma-1)}{\gamma^2-1} (n_y \otimes \vec{\delta}_{N_q, l}) \otimes a \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&+ \left\{ \frac{\partial(\mathcal{F}_1^\pm)}{\partial v_l} \otimes \left[\frac{2}{\gamma^2-1} a \otimes a \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{int}, N_{int}, r} \rangle \\
&+ \left\{ \frac{(\partial \mathcal{F}_1^\pm)}{\partial v_l} \otimes \left[\frac{u \otimes u + v \otimes v}{2} \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_q, N_q, r} \rangle \\
&+ \left\{ \mathcal{F}_1^\pm \otimes \left[\frac{\vec{\delta}_{N_q, l} \otimes v + v \otimes \vec{\delta}_{N_q, l}}{2} \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_q, N_q, r} \rangle \\
\frac{\partial \mathcal{F}_{4,r}^\pm}{\partial p_l} &= \left\{ \frac{\partial(\mathcal{F}_1^\pm)}{\partial p_l} \otimes \left[-\frac{\gamma-1}{\gamma^2-1} \bar{u} \otimes \bar{u} \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_q, N_{g2}, N_q, r} \rangle \\
&\pm \left\{ \frac{\partial(\mathcal{F}_1^\pm)}{\partial p_l} \otimes \left[\frac{2(\gamma-1)}{\gamma^2-1} \bar{u} \otimes a \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&\pm \left\{ (\mathcal{F}_1^\pm) \otimes \left[\frac{2(\gamma-1)}{\gamma^2-1} \bar{u} \otimes \frac{\partial a}{\partial p_l} \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{g2}, N_{int}, r} \rangle \\
&+ \left\{ \frac{\partial(\mathcal{F}_1^\pm)}{\partial p_l} \otimes \left[\frac{2}{\gamma^2-1} a \otimes a \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{int}, N_{int}, r} \rangle \\
&+ \left\{ \mathcal{F}_1^\pm \otimes \left[\frac{2}{\gamma^2-1} \frac{\partial a}{\partial p_l} \otimes a \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{int}, N_{int}, r} \rangle \\
&+ \left\{ \mathcal{F}_1^\pm \otimes \left[\frac{2}{\gamma^2-1} a \otimes \frac{\partial a}{\partial p_l} \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_{int}, N_{int}, r} \rangle \\
&+ \left\{ \frac{(\partial \mathcal{F}_1^\pm)}{\partial p_l} \otimes \left[\frac{u \otimes u + v \otimes v}{2} \right] \right\} \cdot \langle \vec{\Psi}_{N_q, N_q, N_q, r} \rangle
\end{aligned} \tag{C.5}$$

It should be noted that finding the derivatives of the terms like $\frac{\partial a}{\partial p_l}$, $\frac{\partial a}{\partial p_l}$, $\frac{\partial(M\pm 1)^2}{\partial p_l}$, $\frac{\partial(M\pm 1)^2}{\partial u_l}$, $\frac{\partial(M\pm 1)^2}{\partial v_l}$, and $\frac{\partial(M\pm 1)^2}{\partial p_l}$ for the Jacobians of the VLSFV involves solving a linear problem

of the form,

$$\frac{\partial \vec{\alpha}}{\partial q_l} = \begin{pmatrix} \Psi_0(\xi_0) & \Psi_1(\xi_0) & \cdots & \Psi_{N_{int}}(\xi_0) \\ \Psi_0(\xi_1) & \Psi_1(\xi_1) & \cdots & \Psi_{N_{int}}(\xi_1) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_0(\xi_{N_{int}}) & \Psi_1(\xi_{N_{int}}) & \cdots & \Psi_{N_{int}}(\xi_{N_{int}}) \end{pmatrix}^{-1} \frac{d}{dq_l} \begin{Bmatrix} r_0 \\ r_1 \\ \vdots \\ r_{N_{int}} \end{Bmatrix} \quad (\text{C.6})$$

where $\vec{\alpha}$ are the deterministic modes of any generic variable for user-input N_{int} modes, $q_l = (\rho_l, u_l, v_l, p_l)$, and \vec{r} is the corresponding right hand side of the generic variable. The definition of \vec{r} is best explained through an example. Suppose one wants to compute the deterministic density inverse modes for a stochastic variable, ρ_{inv}^* , then

$$\alpha^* = \frac{1}{\rho^*}$$

$$\begin{pmatrix} \Psi_0(\xi_0) & \Psi_1(\xi_0) & \cdots & \Psi_{N_{int}}(\xi_0) \\ \Psi_0(\xi_1) & \Psi_1(\xi_1) & \cdots & \Psi_{N_{int}}(\xi_1) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_0(\xi_{N_{int}}) & \Psi_1(\xi_{N_{int}}) & \cdots & \Psi_{N_{int}}(\xi_{N_{int}}) \end{pmatrix} \begin{Bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{N_{int}} \end{Bmatrix} =$$

$$\begin{Bmatrix} \frac{1}{\rho_0 \Psi_0(\xi_0)} + \frac{1}{\rho_1 \Psi_1(\xi_0)} + \cdots + \frac{1}{\rho_{N_q} \Psi_{N_q}(\xi_0)} \\ \frac{1}{\rho_0 \Psi_0(\xi_1)} + \frac{1}{\rho_1 \Psi_1(\xi_1)} + \cdots + \frac{1}{\rho_{N_q} \Psi_{N_q}(\xi_1)} \\ \vdots \\ \frac{1}{\rho_0 \Psi_0(\xi_{N_{int}})} + \frac{1}{\rho_1 \Psi_1(\xi_{N_{int}})} + \cdots + \frac{1}{\rho_{N_q} \Psi_{N_q}(\xi_{N_{int}})} \end{Bmatrix}$$

where,

$$\vec{r} = \begin{Bmatrix} \frac{1}{\rho_0 \Psi_0(\xi_0)} + \frac{1}{\rho_1 \Psi_1(\xi_0)} + \cdots + \frac{1}{\rho_{N_q} \Psi_{N_q}(\xi_0)} \\ \frac{1}{\rho_0 \Psi_0(\xi_1)} + \frac{1}{\rho_1 \Psi_1(\xi_1)} + \cdots + \frac{1}{\rho_{N_q} \Psi_{N_q}(\xi_1)} \\ \vdots \\ \frac{1}{\rho_0 \Psi_0(\xi_{N_{int}})} + \frac{1}{\rho_1 \Psi_1(\xi_{N_{int}})} + \cdots + \frac{1}{\rho_{N_q} \Psi_{N_q}(\xi_{N_{int}})} \end{Bmatrix}$$